Universidad Carlos III de Madrid.

Diseño de algoritmo para la detección de adelantamientos de vehículos en plataforma Android.

Grado en Ingeniería Electrónica Industrial y Automática

AUTOR: María Isabel García Zamora

TUTOR: Fernando García Fernández

Leganés, Septiembre de 2015.

Título: Diseño de algoritmo para la detección de adelantamientos de vehículos en plataforma Android.

Autor: María Isabel García Zamora

Tutor: Fernando García Fernández

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario:_____

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día __ de Octubre de 2015 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

A Dios por todas las bendiciones que me ha concedido.

A mí familia y amigos por todo el apoyo recibido durante estos meses de trabajo.

Resumen

El presente proyecto consiste en el estudio de la posibilidad de crear una aplicación Android con el objetivo de detectar si el vehículo en el cual se circula es adelantado.

Para ello se utilizará un dispositivo Android con cámara incorporada y un algoritmo de flujo óptico que permita obtener vectores de movimiento. Dicho algoritmo es el algoritmo de Lucas-Kanade que permite identificar regiones de una imagen cuyo movimiento es similar y por tanto, mediante dos fotogramas consecutivos, obtener el movimiento de estas regiones, consiguiendo así vectores de movimiento.

Como herramienta, se dispone de una tablet Nexus 10, cuya cámara es de 5 MP, tiene una CPU A15 de doble núcleo, 2GB de RAM y cuyo sistema operativo es el Android 5.1.1.

El objetivo del proyecto es crear una aplicación Android que ayude a mejorar la seguridad vial, disminuyendo los accidentes por colisiones laterales, que son la tercera causa de muerte en carretera en España; además de crear un sistema de seguridad sencillo y económico.

Palabras clave: flujo óptico, Lucas-Kanade, movimiento, Android, aplicación, detección de adelantamientos y seguridad vial.

Abstract

This Project consists in the study of the possibility of creating an Android application in order to detect if the vehicle in which it is being circulated, it is being overtaken.

For this propose, an Android device, which has an incorporated camera, will be used. Besides, it will be used an optical flow algorithm which allows to obtain vectors of movement. This algorithm is the Lucas-Kanade algorithm which allows identifying regions of an image whose movement is similar and therefore, thanks to two consecutive frames, it can be obtained the movement of these regions; moreover it can be obtained vectors of movement.

In order to develop this project, it is used a Nexus 10 which is a tablet whose camera is of 5MP, it has an A15 dual-core CPU, 2GB of RAM and whose operating system is Android 5.1.1.

The aim of the project is to create an Android application in order to improve the road safety, reducing accidents of lateral collisions, which are the third leading cause of death in road in Spain; besides another aim is to create a simple and economic system of safety.

Keywords: optical flow, Lucas-Kanade, movement, Android, application, overtaking's detection and road safety.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1
Introducción

Actualmente, el uso de los vehículos se ha convertido en algo imprescindible en los últimos tiempos en todo el mundo. Estos son usados continuamente tanto en situaciones cotidianas como ir a trabajar, ir a llevar a los niños al colegio o ir de compras, hasta situaciones esporádicas como hacer turismo o salir de vacaciones. Una muestra de ello es que en los últimos 27 años se ha duplicado el número de vehículos en España, pasando de 303 vehículos por cada 1000 habitantes en el año 1985 a 675 en 2012, incluso llegando a la cifra de 685 en 2007.

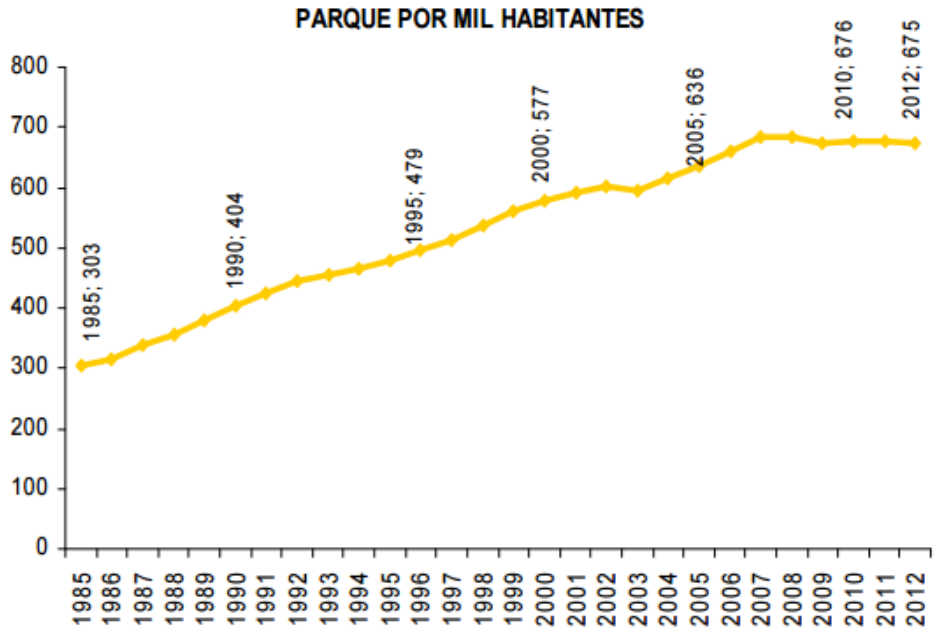

Figura 1.1. Relación entre población y parque de vehículos en España desde 1985 hasta 2012. Fuente cifras de población: INE. Estimaciones de la población de España a 1 de enero de 2012. (Fuente utilizada por EUROSTAT).

Pero como toda herramienta siempre hay ciertos inconvenientes, y el caso de los vehículos no es una excepción. Uno de los mayores es el número de accidentes de tráfico.

La evolución de las cifras de fallecidos por accidente de tráfico con víctimas muestra periodos diferenciados: en las décadas de los años 60, 70 y 80 se observa un incremento, de forma que en el año 1989 se produjo el pico de mortalidad por accidente de tráfico en España con 9.344 fallecidos. A partir de los 90 se observa un periodo de descenso, de forma que en cuatro años se produce una reducción del 33%. Entre 1995 y 2003 no se observa una tendencia definida (±5%) y a partir del año 2004 hasta el 2013 se vuelve a producir un periodo de descenso continuado. [1]

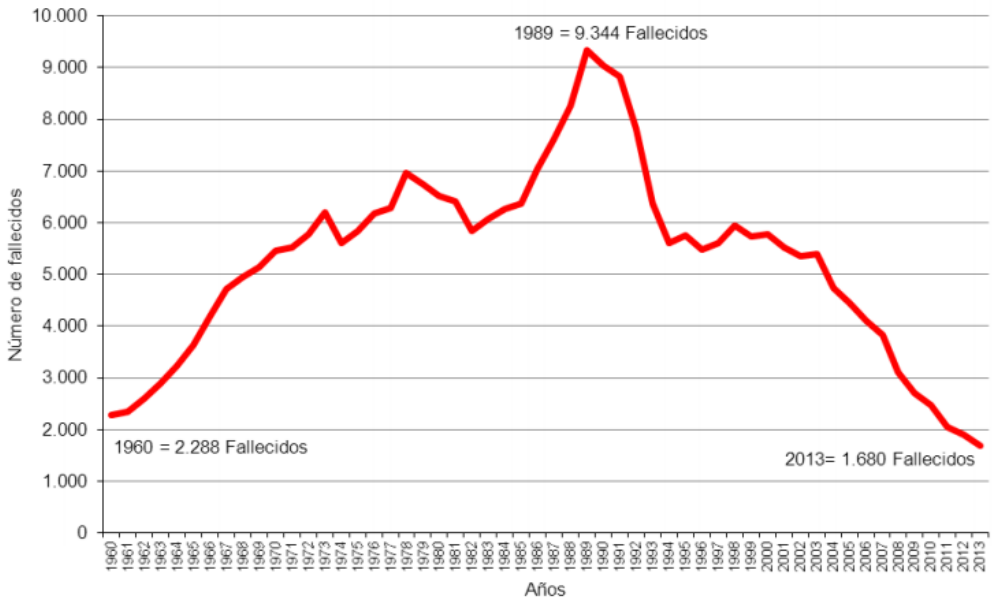

Figura 1.2. Muertos por accidentes de tráfico desde 1965 hasta 2013

La evolución de fallecidos, heridos graves y heridos leves ha variado desde 1965 a 2013. En 1965 las proporciones eran 5% fallecidos, 26% heridos graves y 68% heridos leves y se mantienen prácticamente hasta 1998, año en el que esa proporción fue 4% fallecidos, 24% heridos graves y 72% heridos leves. En 2003 la proporción cambia reduciéndose la de fallecidos al 3% y la de heridos graves al 17% y vuelve a cambiar a partir de 2004 descendiendo la proporción de fallecidos y de heridos graves hasta 2013 en el que ha sido del 1,3% y del 8,0% respectivamente. [1]

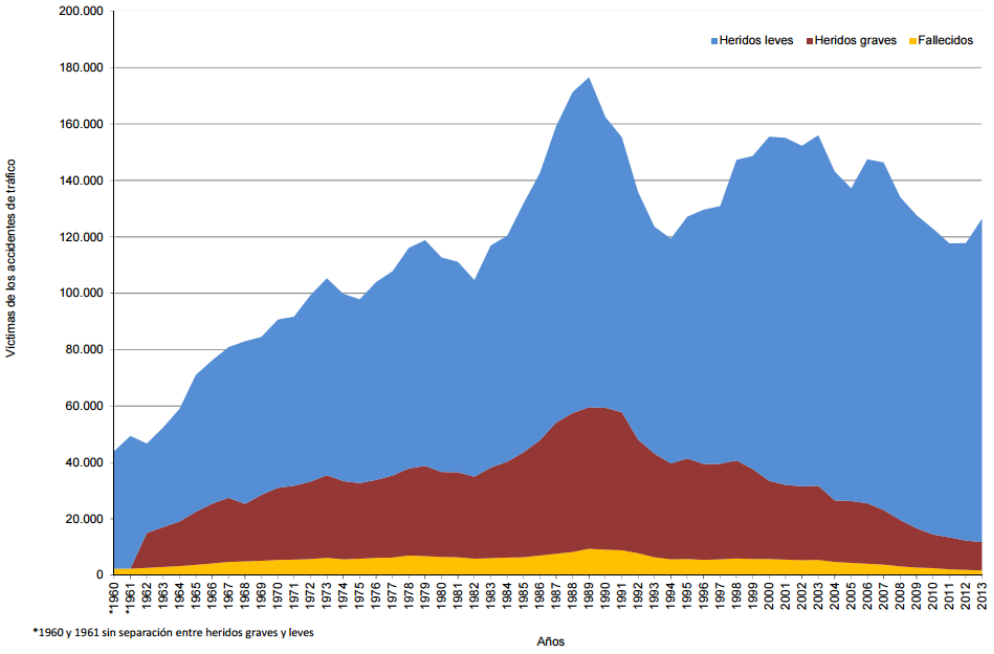

Figura 1.3. Víctimas de los accidentes de tráfico desde 1965 hasta 2013.

Por tanto, como muestran los datos, los accidentes de tráfico son uno de los mayores problemas que tiene el uso de vehículos, por lo que aumentar la seguridad es una prioridad.

Para ello, se ha seguido una línea de investigación y desarrollo de nuevas tecnologías que permitan aumentar la seguridad. En un principio, los esfuerzos se concentraron en minimizar los daños causados por los accidentes de tráfico, dando lugar a los sistemas de seguridad pasiva como los cinturones de seguridad, carrocerías que absorban la energía del accidente, el airbag... Sin embargo, a partir, de los años 70, gracias a la incursión de la electrónica y de los sistemas informáticos, los esfuerzos se redirigieron hacia la creación de sistemas que evitaran los accidentes, dando lugar a sistemas como el ABS o el ESP, originando los sistemas de seguridad activa. Algunos de estos novedosos sistemas son de uso obligatorio.

Actualmente, las compañías investigan nuevos sistemas que permitan seguir reduciendo el número de accidentes, así como mejorar el confort y las capacidades del conductor. Algunos de estos nuevos sistemas son: el sistema de detección de ángulo muerto, de aviso de cambio involuntario de carril, de visión nocturna, de control y alerta de sueño, de detección de señales y peatones y control de velocidad adaptativo.

Por otro lado, el uso de los dispositivos Android ha crecido en los últimos años. A finales de 2013, gracias a la expansión de este sistema a más sectores, la irrupción de los Android low-cost y el tímido avance de la competencia, Android alcanzó la cifra de 1,9 mil millones dispositivos, con un crecimiento de 877 mil millones solo en 2013, un crecimiento que permitió al sistema de Google estar presente en el 38% de los dispositivos actuales, una cifra que preveían que aumentara al 45% en 2014, es decir, aproximadamente una de cada seis personas en la tierra tendrían un dispositivo Android. [2]

Worldwide Device Shipments by Operating System (Thousands of Units)

Operating System	2012	2013	2014	2015
Android	503,690	877,885	1,102,572	1,254,367
Windows	346,272	327,956	359,855	422,726
iOS/Mac OS	213,690	266,769	344,206	397,234
RIM	34,581	24,019	15,416	10,597
Chrome	185	1,841	4,793	8,000
Others	1,117,905	801,932	647,572	528,755
Total	**2,216,322**	**2,300,402**	**2,474,414**	**2,621,678**

Source: Gartner (December 2013)

Figura 1.4. Estadística sobre el uso de Android en 2013 y previsión para el 2014 y 2015.

Para 2014 se preveía que Android estuviera en todas partes y aumentara su número en otros 1,1 mil millones más. De las tablets se vendieron 263 millones y de dispositivos híbridos y demás hubo un salto hasta los 39 millones, una cifra mucho más baja que la de smartphones pero que ayuda a entender como los portátiles y PCs bajaron sus cifras de los 300 a los 277 millones; por tanto se vendían casi el triple de Android que PCs. [2]

Aunque los datos son de 2013, el uso del sistema Android no ha disminuido.

La facilidad con la que las aplicaciones Android son creadas y puestas en el mercado y la posibilidad de hacer negocio con ellas, hacen que Android esté presente en todos los ámbitos de la vida.

1.1. Objetivo del trabajo

Como demuestran las estadísticas, los accidentes de tráfico siguen siendo una constante a pesar de las muchas campañas de concienciación y de las mejoras en la seguridad.

Por otro lado, los dispositivos Android son una herramienta cotidiana en nuestro día.

Por tanto, este trabajo surge de la unión entre la necesidad de creación y/o mejora de sistemas de seguridad en los vehículos y el auge de los dispositivos Android.

En resumen, el objetivo es explorar la posibilidad de que se pueda usar un dispositivo Android para detectar si el vehículo en el cual se circula es adelantado, para lo cual se usará la cámara del dispositivo Android y un algoritmo de flujo óptico capaz de detectar el movimiento del vehículo que adelanta.

Este trabajo puede servir de base a nuevos proyectos que ayuden a la seguridad vial y también potenciar los sistemas Android llevándolos a su límite.

Como objetivos principales cabe destacar los siguientes:

- Sentar las bases para una nueva línea de investigación que descubra si los dispositivos Android son viables para la seguridad vial.
- Ayudar a mejorar la seguridad vial, disminuyendo todo lo posible los accidentes por colisiones laterales que son la tercera causa de muerte en carretera en España.
- Ofrecer la posibilidad de crear un sistema de detección de adelantamientos sencilla y económica.

1.2. Estructura de la memoria

La presente memoria se compone de 7 capítulos en los cuales se irán desgranando todos los detalles del trabajo.

El primer capítulo está formado por una introducción a los datos estadísticos de accidentes de tráfico, al uso de dispositivos Android, así como el objetivo del trabajo y una breve descripción de la estructura del documento. En el segundo capítulo se explicarán los principales sistemas de seguridad activa presentes en los vehículos. En el tercer capítulo se dará una explicación sobre las herramientas tanto hardware como software que han tomado parte en este proyecto. En el cuarto capítulo se expondrán los distintos pasos que se han ido siguiendo para su elaboración. En el quinto capítulo se detallarán los resultados obtenidos. En el sexto capítulo, se expondrán las conclusiones a las que se han llegado y los posibles trabajos futuros. En el séptimo capítulo se presentará un presupuesto del trabajo. El capítulo octavo detalla las

referencias utilizadas. Y finalmente, se adjunta un anexo con una serie de tablas obtenidas en el transcurso del trabajo.

Capítulo 2
Estado del Arte

Entre los sistemas de seguridad cabe destacar el ABS (antibloqueo de frenos) y el ESP (control de estabilidad) que son sistemas de seguridad activa de indudables beneficios para la conducción, que aunque siguen evolucionando, no son novedosos. Ahora los novísimos son los sistemas de detección del entorno, entre los cuales están los siguientes.

2.1. Sistema de detección de sueño

Puede tener varios nombres (por ejemplo detección de fatiga) pero en el fondo las diferentes versiones consisten en lo mismo, intentar detectar si el conductor no está en óptimas condiciones para seguir conduciendo.

Normalmente son un sistema electrónico con un sensor en el volante, que cuenta cuántas veces por minuto el conductor realiza pequeñas correcciones en la dirección. Se sabe que para mantenernos en el carril, los conductores no mantenemos el volante quieto y fijo, sino que corregimos casi constantemente dos o tres grados hacia la derecha o hacia la izquierda, para intentar ir lo más centrados posible en él.

Si el pequeño procesador del sistema cuenta menos correcciones por minuto de lo que se considera normal, interpreta que el conductor puede estar distraído, estar cansado o incluso estar durmiéndose al volante, así que advierte de ello al conductor. Esta advertencia puede ser variable; lo normal es un cartel en la pantalla digital del cuadro de instrumentos y una alarma sonora (por ejemplo un pitido) pero también puede ser incluso una vibración en el volante.

El objetivo es evitar que un conductor se duerma al volante sin darse cuenta.

Figura 2.1. Sistema de detección del sueño

De manera más experimental se están probando sistemas de reconocimiento facial (mediante una pequeña cámara) que detecten si un conductor está cerrando los ojos más de la cuenta y se está durmiendo. [3]

2.2. Sistema de detección de cambio de carril involuntario

Si en algún momento pisamos o sobrepasamos alguna de las líneas, y no se había activado el intermitente (luz indicadora de cambio dirección) del lado correspondiente, el sistema interpreta que el cambio es involuntario (por un despiste) y actúa. Algunos sistemas solo advierten al conductor, normalmente con una vibración en el asiento o bien en el volante, pero también puede sonar una pequeña alarma sonora, y también un mensaje en la pantalla digital del cuadro de instrumentos.

Otros sistemas un poco más avanzados, actúan sobre la dirección, y hacen girar el volante ligeramente en la dirección opuesta, para corregir la deriva en la trayectoria que hace que pisemos la línea en cuestión. Cada vez más sistemas de dirección asistida utilizan un servomotor eléctrico, así que no es complicado utilizar este motor para hacer girar el volante.

Estos sistemas, por el momento, no mantienen el coche en el carril, solo corrigen un poco el volante. Por supuesto funcionan también de noche, o con visibilidad reducida por niebla (eso sí, a partir de unos 50 o 60 km/h de velocidad). [3]

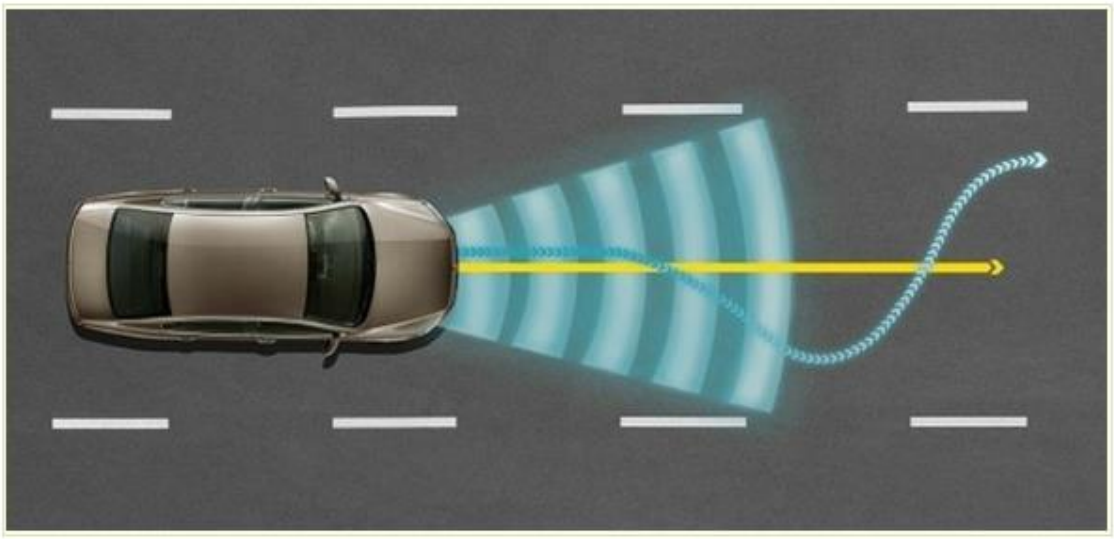

Figura 2.2. Sistema de detección de cambio de carril involuntario

2.3. Advertencia de ángulo muerto en espejo retrovisor

El ángulo muerto en los espejos retrovisores exteriores siempre ha estado ahí (es esa pequeña área del campo visual hacia atrás, muy próxima a nuestro coche, que el espejo no es capaz de reflejar). Sin duda lo mejor es girar ligeramente la cabeza para mirar de reojo y cerciorarse de que no hay ningún vehículo oculto en ese espacio.

Con los años salieron espejos ligeramente curvados para aumentar el ángulo de visión (aunque sigue quedando todavía un poco de espacio oculto) o incluso algunos conductores montan pequeños espejos complementarios que se orienten hacia el ángulo muerto. Hoy en día la tecnología electrónica ya ha previsto una solución.

Algunos coches montan (como opción) un sistema que avisa de vehículos en el ángulo muerto, mediante una señal visual sobre el espejo retrovisor, o en el marco del espejo (suele ser un led amarillo que parpadea, o un pequeño triángulo de peligro). El sistema emplea radares (es decir un sistema de emisión de ondas de radio), en las esquinas del paragolpes trasero (o en el lateral del coche, a veces en el mismo espejo retrovisor) orientados hacia el área que queda oculta. [3]

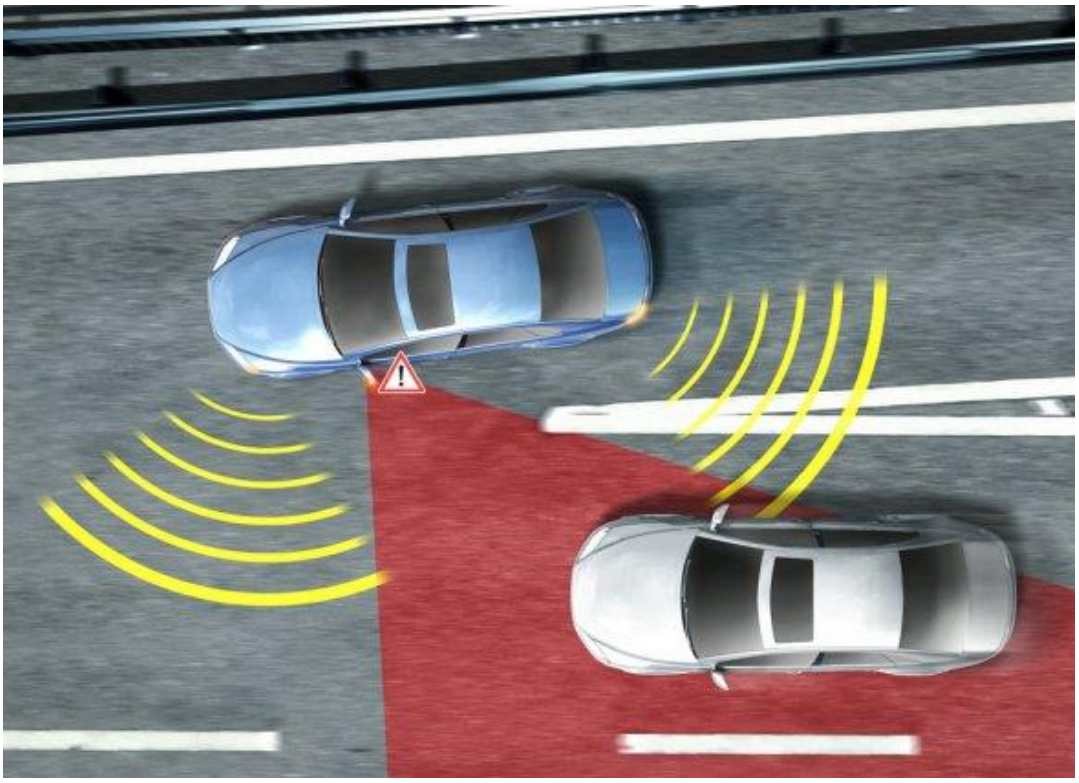

Figura 2.3. Sistema de detección de ángulo muerto

2.4. Reconocimiento de señales de velocidad y adelantamiento

Este sistema, cada vez más presente en los vehículos, incorpora una cámara en la parte alta del parabrisas, junto al espejo retrovisor, que va reconociendo las señales circulares de velocidad máxima, prohibido adelantar, fin de velocidad máxima y fin de prohibido adelantar.

En la pantalla digital del cuadro de instrumentos aparece la velocidad límite del tramo en el que se circula. Se actualiza en tiempo real, de manera que en todo momento el conductor puede saber la velocidad límite del tramo por el que circula. [4]

2.5. Advertencia de sentido contrario

Cada año se dan muchos casos de conductores que, por despiste (por ejemplo por estrés o mala visibilidad), o sobre todo por no estar en condiciones para conducir, circulan en sentido contrario a la marcha, en autovías y autopistas (en Alemania por ejemplo se dan unos 1.800 casos al año).

Un ejemplo de sistema de advertencia de sentido contrario es el sistema Mercedes Real Life Assist que basa su capacidad en el sistema de detección y lectura de señales que incorporan los modelos de la marca. Pero no solo usa este sistema, sino que ahora esta tecnología es capaz de reconocer el momento en el que el vehículo se está adentrando en un carril de sentido opuesto gracias a una cámara de vídeo situada en el parabrisas y al sistema de navegación GPS.

El sistema, al detectar que la señal de sentido contrario a la que el vehículo se aproxima es ignorada, lleva a cabo un proceso de alerta visual en el cuadro de instrumentos y acústica en el interior para informar al conductor y a los posibles pasajeros.

Gracias a su implantación, Mercedes pretende mitigar este tipo de situaciones cuando son producidas por equivocaciones o despistes durante la conducción. [5]

2.6. Control de velocidad

El control de velocidad permite una conducción más cómoda y relajada al no tener que ir pendiente del velocímetro por si excede la velocidad máxima permitida.

Se puede seleccionar la velocidad de circulación que se desee, y el procesador se encargará de gestionarla para circular de manera constante a la velocidad que hubiésemos programado.

Si durante la conducción tuviésemos que reducir la velocidad, el sistema se desactiva al pisar el freno; de igual manera sucede si tuviésemos que acelerar, el coche incrementa su velocidad hasta que dejemos de accionar el acelerador y el vehículo vuelve a la velocidad que teníamos programada.

Una evolución de este sistema es el control de velocidad adaptativo, que mantiene la distancia de seguridad adecuada con el vehículo que nos precede. Si se estuviera demasiado cerca, el sistema decelera para aumentar la distancia de seguridad.

Y, si es preciso, también puede actuar sobre el freno para reducir la velocidad de forma más rápida y conseguir la distancia de seguridad necesaria. [4]

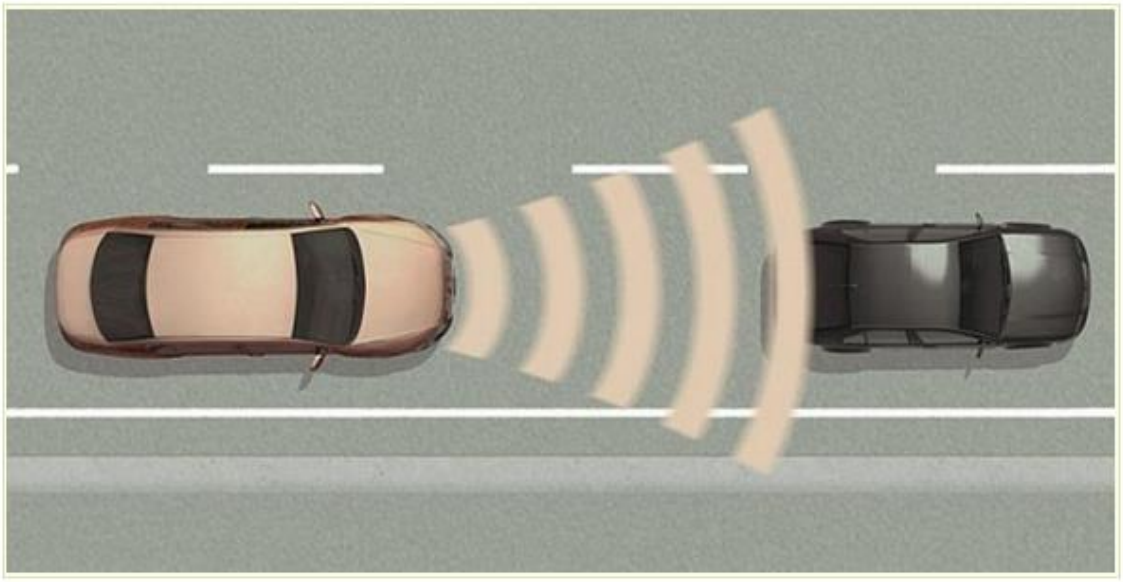

Figura 2.4. Sistema de control de velocidad adaptativo

2.7. Reconocimiento de objetos

Este sistema pretende evitar sobre todo atropellos de peatones o ciclistas. Es similar al sistema de reconocimiento de peatones nocturno. El sistema ContiGuard quiere ir un paso más allá y por el momento es algo experimental. Este sistema consiste en dos cámaras de alta resolución, colocadas en la parte alta del parabrisas, que permiten tener una visión estereográfica al microprocesador del sistema.

Figura 2.5. Sistema de reconocimiento de objetos

El sistema es capaz de distinguir entre peatones, ciclistas, coches y otros objetos, medir la distancia hasta ellos, y predecir su trayectoria, verificando si esta se corta con la que lleva el coche, y determinar por tanto si existe riesgo de un accidente, para tomar las medidas preventivas conducentes a evitarlo o aminorar las consecuencias. Advertirá al conductor, e incluso podría accionar automáticamente los frenos del vehículo.

Otro sistema que se está probando (en Alemania, bajo el nombre Amulett Car2X) funciona por ondas de radio. Los peatones y ciclistas deberían llevar un pequeño transpondedor, muy similar al sistema RFID, que incluso no necesitaría alimentación eléctrica (si es de tipo pasivo).

El coche debe llevar un emisor-receptor y recibe la señal de respuesta de los transpondedores. Lo bueno de este sistema es que los peatones podrían estar ocultos (por ejemplo niños detrás de un vehículo alto estacionado) pero el microprocesador del coche sabría que están ahí y advertiría de ello al conductor. [3]

2.8. Sistemas anticolisión

Ya hemos visto que con cámaras o radares, el "cerebro" electrónico del coche puede saber que hay obstáculos delante de nuestra trayectoria, ya sea un peatón, ya sea otro vehículo. Así que si determina que existe un riesgo real de atropello o colisión, puede reaccionar antes incluso que el conductor y accionar los frenos automáticamente para reducir la velocidad, o incluso detener por completo el coche y evitar el atropello o choque.

Normalmente los sistemas de frenado automático hasta la detención, suelen funcionar a bajas velocidades (lo habitual es que sea por debajo de 40 o 50 km/h, para ciudad o atascos de tráfico, y evitar un despiste). Algunos sistemas solo alertan al conductor y preparan el sistema de frenado para que actúe con la máxima fuerza de frenado, mientras que otros también frenan ellos mismos el coche. Volvo (sistema city safety), y muchas otras marcas, disponen ya de sistemas de este tipo.

Similar a este es el sistema, por ahora todavía experimental, de anticipación de curvas de BMW Foresight Transmission Control. En combinación con el GPS del coche, reconoce cuándo el vehículo está llegando a una curva, reconoce la dificultad de la misma y estima la velocidad óptima de trazado de la misma con seguridad. Si la velocidad no es adecuada actúa automáticamente sobre los frenos para reducir la velocidad, y reduce una marcha en la caja de cambios. [3]

2.9. Sistema de comunicación entre coches para aviso y alerta

Los coches podrían estar equipados con un sistema de comunicación local inalámbrica, con un alcance de unos metros (500 o 600 m) de modo que podrían transmitirse mensajes entre los coches, por proximidad, e ir retransmitiendo el mensaje unos a otros.

Es por ejemplo el sistema Ford CoCarX en pruebas. Imaginemos que un coche sufre un accidente en mitad de una autovía, y obstaculiza la calzada, suponiendo un peligro para el resto de coches que vengan detrás, el coche accidentado manda inmediatamente el mensaje de aviso y lo reciben los coches cercanos que se van acercando a él.

El conductor visualiza el mensaje de alerta en la pantalla digital del cuadro o de la consola central, y está advertido de que hay problemas unos metros más adelante, pudiendo estar prevenido, aminorar la marcha, y evitar cualquier peligro, por ejemplo, por alcance al coche que está accidentado o averiado.

El sistema incluso puede hacer que todos los otros sistemas de seguridad estén alerta, y actúen de manera automática, antes incluso de que el conductor asimile el mensaje de alerta y reaccione. [3]

Figura 2.6. Ejemplo de uso del sistema de comunicación entre coches

Capítulo 3
Descripción de la Arquitectura

El desarrollo del sistema de detección de adelantamientos requiere un dispositivo Android con cámara, un software de desarrollo y una librería de visión artificial. Como elementos secundarios, es necesario un ordenador y un monitor. A continuación se exponen las distintas herramientas utilizadas.

3.1. Arquitectura hardware

3.1.1. Tablet Nexus 10

Características de interés para el proyecto [6]

- ⇒ Cámara de 5 MP (principal), grabación de vídeo a 1080p
- ⇒ CPU: A15 de doble de núcleo
- ⇒ GPU: Mali T604 de cuatro núcleos
- ⇒ RAM: 2GB
- ⇒ Conectividad: Micro USB
- ⇒ Sistema operativo: Android 5.1.1

Figura 3.1. Tablet Nexus 10

3.1.2. Ordenador HP

- Modelo: HP Pavilion dm3 Notebook PC.
- Procesador: AMD Athlon(tm) Neo X2 Dual Core Processor L335 1.60 GHz.
- Memoria instalada (RAM): 2,00 GB (1,75 GB utilizable).
- Tipo de sistema: Sistema operativo de 64 bits.

3.1.3. Monitor auxiliar

- Resolución de 1024 x 768

3.2. Arquitectura software

3.2.1. Eclipse Juno

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. [7]

Eclipse Juno fue lanzado en 2012, siendo la versión 4.2 del software.

Por otra parte, el Android Development Tool (ADT) es un plugin para el IDE Eclipse que está diseñado para dar un potente entorno integrado en el que poder desarrollar aplicaciones de Android.

ADT amplía las capacidades de Eclipse para que pueda configurar rápidamente nuevos proyectos para Android, crear una interfaz de usuario de la aplicación, agregar paquetes basados en la API de Android Framework, depurar sus aplicaciones utilizando las herramientas del SDK de Android, e incluso exportar archivos .apk con el fin de distribuir la aplicación. [8]

3.2.2. Librería de tratamiento de imagen OpenCv 2.4.11

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

OpenCV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estérea y visión robótica. [9]

En conjunto con la librería de OpenCv que hay que incorporar al proyecto en eclipse, también es necesaria la instalación del OpenCV Manager en el dispositivo Android, que permite la ejecución de aplicaciones OpenCv en el dispositivo. La versión que se ha usado de esta aplicación es la 2.4.9.0 rev 1.

3.3. Lenguaje de programación: Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos 10 millones de usuarios reportados. [10]

A pesar de que Eclipse permite la programación en C/C++ (NDK: Native Development Kit), y en su origen las funciones proporcionadas por la librería de OpenCv están en C/C++, se decidió programar en Java, ya que los algoritmos usados de OpenCv habían sido adaptados a este lenguaje, además de que en Android es mucho más sencillo programar en Java que en NDK, sobre todo a la hora de compilar el código.

Capítulo 4
Sistema propuesto

El sistema propuesto se basa en un algoritmo de reconocimiento de movimiento, de tal forma que si hay n vectores de movimientos válidos (con unas características específicas), el sistema indica que se está produciendo un adelantamiento. El algoritmo en cuestión es el de Lucas-Kanade que se puede encontrar en la librería de OpenCv.

El diagrama de flujo del sistema propuesto es el siguiente:

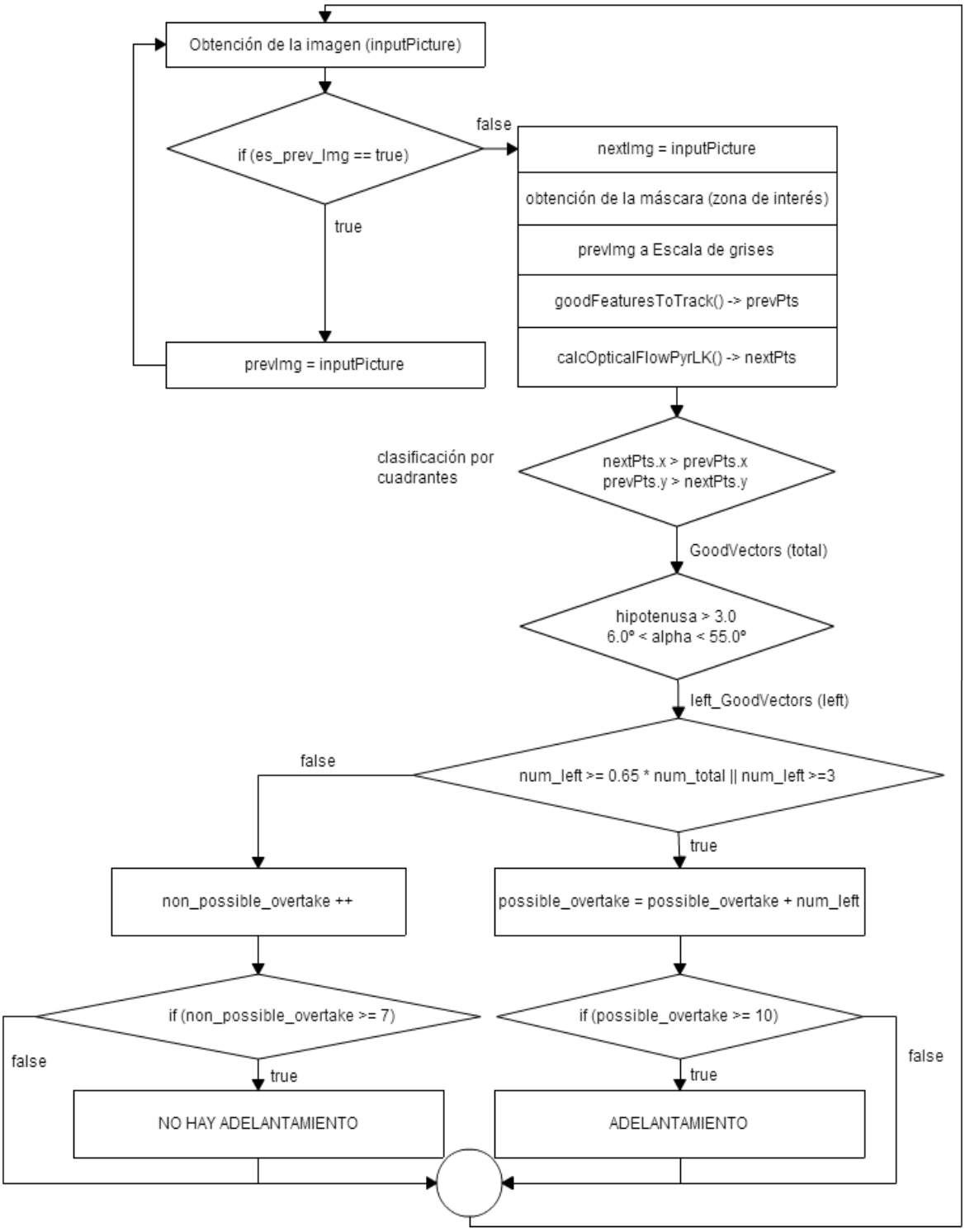

Figura 4.1. Diagrama de flujo del sistema propuesto

4.1. Obtención de imágenes a través de la cámara del dispositivo Android

Para obtener las imágenes de la cámara Android se ha usado la librería Android de OpenCv que se vincula con el layout a través de CameraBridgeViewBase.

A pesar de que la tablet permitía una resolución de 1920x1080, se decidió usar una resolución menor pues de esa forma se daba mayor fluidez a la aplicación. Tras varias pruebas, la resolución escogida fue la de 640x480, debido a que si era menor, apenas se detectaban puntos de interés y si era mayor, la velocidad de captura de imágenes era muy lenta, porque cuanta mayor es la resolución, mayor es la cantidad de píxeles que hay que procesar.

En esta parte del sistema, cabe destacar dos funciones: void onCameraViewStarted(int width, int height) y Mat onCameraFrame(CvCameraViewFrame inputFrame).

La función void onCameraViewStarted(int width, int height) es llamada la primera vez que se ejecuta la aplicación y permite configurar la altura y el ancho de las variables que se usarán para albergar los dos fotogramas y la imagen final.

Por otra parte, la función Mat onCameraFrame(CvCameraViewFrame inputFrame) permite obtener un fotograma de la cámara y mostrarlo por pantalla. Este fotograma es el input de la función Mat LK_ Frame(Mat inputPicture) que pertenece a la clase Optical_Flow, que es la clase donde están las funciones que hacen los cálculos y los procedimientos para discernir si existe un adelantamiento. Para poder pasar de CvCameraViewFrame a Mat usamos rgba().

Pero el flujo de imágenes no es 1 y 2, 2 y 3, 3 y 4…etc., sino que es 1 y 2, 3 y 4, 5 y 6…etc. Lo lógico es hacerlo como en la primera serie, pero después de varias pruebas, al no dejarle a la tablet tiempo de procesamiento suficiente entre fotogramas, la velocidad disminuía y las lecturas no eran nada fiables, por lo que finalmente se optó por la segunda serie.

Dicho esto, el funcionamiento es el siguiente: se obtiene una imagen y se guarda; se obtiene una segunda imagen y se guarda; se procesan ambas imágenes y se devuelve la imagen final con los vectores de movimiento de dibujados. Dicha imagen es la que se muestra por pantalla. El tiempo entre la obtención de la primera y la segunda imagen es mínimo por lo que el ojo no detecta que la función LK_Frame() no está devolviendo nada a onCameraFrame() y por tanto no se muestra nada por pantalla. Sin embargo esto permite que haya tiempo suficiente para que la tablet no se colapse y pueda tener un flujo constante y medianamente fluido de imágenes.

4.2. GoodFeaturesToTrack(): obtención de puntos de interés en la imagen

La función goodFeaturesToTrack permite obtener un vector que representa las coordenadas de las esquinas más significativas de una imagen o de una región de la imagen.

La función calcula la calidad de las esquinas en cada píxel de la imagen. A continuación, la función realiza una supresión no-máxima (se conservan los máximos locales en la vecindad de 3x3). Las esquinas con el valor mínimo menor que:

$$qualityLevel \cdot max_{x,y} qualityMeasureMap(x,y)$$

son rechazados. Las esquinas que quedan son ordenadas por la medida de la calidad en orden descendente. Finalmente, la función devuelve aquellas esquinas que están a una distancia mayor que $minDistance$ unas de otras. [11]

4.2.1. Aplicación de la función goodFeaturesToTrack()

Una vez obtenidas ambas imágenes (prevImg y nextImg), obtenemos las dimensiones de estas, para poder conseguir una zona de detección, es decir, una máscara que permite a la función goodFeaturesToTrack() centrarse en un área específica de la imagen.

La necesidad de esta máscara es porque realmente no se necesita más información que la de la parte izquierda media de la imagen:

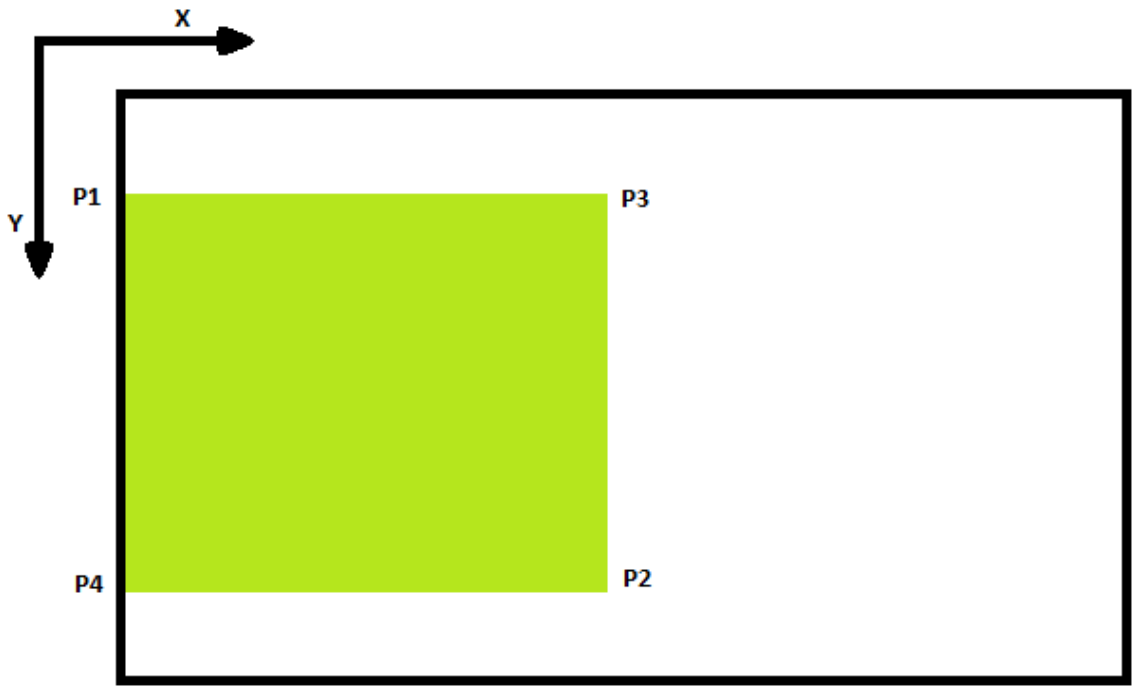

Figura 4.2. Región de interés en la imagen (zona verde)

A efectos prácticos, la máscara además de hacer que la función solo se centre en la zona de interés, también hace que la función tenga menos píxeles que procesar y por tanto la velocidad de procesamiento es mayor. Como en el anterior apartado, la fluidez de las imágenes tiene que ser el óptimo; en caso contrario la aplicación iría muy lenta.

Una vez obtenidos los puntos P1, P2, P3 y P4, la aplicación llama a la función calculation_LK() que es la que realiza los cálculos del flujo óptico.

En primer lugar, procesamos la imagen prevImg y la pasamos a escala de grises para que la localización de puntos por parte de la función goodFeaturesToTrack() sea efectiva. Tras este proceso se obtiene la imagen llamada scene.

En segundo lugar, creamos la máscara gracias a los puntos de las esquinas. Con ellos se crea una lista de MatOfPoint que contiene todos los puntos que encierra el rectángulo formado por estos 4 puntos (esta lista solo se hace la primera vez). Una vez obtenidos todos los puntos, estos se pasan a la variable Mat llamada mask mediante el procedimiento drawContours() de la librería Imgproc.

En este punto ya se ha obtenido la máscara y la imagen ya está lista para que la función goodFeaturesToTrack sea aplicada.

```
goodFeaturesToTrack(
        image => scene
        corners  => initial_prevPts
        maxCorners => 30
        qualityLevel => 0.01
        minDistance => 5
        mask => mask
        blockSize => 3
        useHarrisDetector => false
        k => 0.04 );
```

El resultado se ve reflejado en el vector initial_prevPts que es el que contiene la información de los puntos de interés.

Por seguridad, si el vector initial_prevPts está vacío porque no se encuentra ningún punto de interés (caso en el que la cámara está tapada), al vector prevPts (que es el que se le pasa a calcOpticalFlowPyrLK() se le pasan los puntos de las esquinas (P1, P2, P3, P4) para que cuando se ejecute la función de Lucas-Kanade, la aplicación no se colapse y se pare; en caso contrario, simplemente se hace una conversión, para pasar los datos desde initial_prevPts hasta prevPts. El motivo de esta conversión es debido a que initial_prevPts es del tipo MatOfPoint y prevPts es del tipo MatOfPoint2f(). La conversión se hace pasando la información initial_prevPts a un array y haciendo que prevPts obtenga su valor desde un array:

$$prevPts.fromArray(initial\_prevPts.toArray());$$

4.3. Lucas-Kanade: obtención de los vectores de movimiento

4.3.1. Base teórica

El flujo óptico es el patrón de movimiento aparente de unos objetos en una imagen entre dos fotogramas consecutivos, causado por el movimiento del objeto o de la cámara. Es un vector de 2 dimensiones que muestra el movimiento de los puntos entre el primer fotograma y el segundo.

El flujo óptico trabaja con varios supuestos:

- La intensidad de los píxeles de un objeto no cambian entre dos fotogramas consecutivos.
- Píxeles vecinos tienen un movimiento similar.

Considera un pixel $I(x, y, t)$ en el primer fotograma. Se mueve una distancia (dx, dy) en el segundo frame después de un tiempo (dt), por lo que si esos píxeles son los mismos y su intensidad no cambia, se puede decir que:

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

Se aplican las series de aproximación de Taylor desde el lado derecho, se quitan los términos comunes y se divide por dt para obtener la siguiente ecuación:

$$f_x u + f_y v + f_t = 0$$

donde:

$$f_x = \frac{\partial f}{\partial x}; \quad f_y = \frac{\partial f}{\partial y} \qquad u = \frac{dx}{dt}; \quad v = \frac{dy}{dt}$$

Esta es la ecuación del Flujo Óptico, donde f_x y f_y son los gradientes de la imagen. f_t es el gradiente a lo largo de tiempo. Pero (u, v) son desconocidos.

Para poder solucionar esta ecuación hay varios métodos, uno de ellos el algoritmo de Lucas-Kanade.

Asumiendo que los pixeles vecinos tienen un movimiento similar, el método de Lucas-Kanade toma una zona de 3x3 alrededor de un punto, por lo que estos 9 puntos tienen un movimiento similar. Podemos encontrar (f_x, f_y, f_t) para estos 9 puntos. Ahora el problema es resolver 9 ecuaciones con dos incógnitas lo que es sobre-determinado. Una solución mejor es obtener el resultado con el método de mínimos cuadrados. La solución final es la siguiente: [12]

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i {f_{x_i}}^2 & \sum_i f_{x_i} f_{y_i} \\ \sum_i f_{x_i} f_{y_i} & \sum_i {f_{y_i}}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i f_{x_i} f_{y_i} \\ -\sum_i f_{x_i} f_{y_i} \end{bmatrix}$$

4.3.2. Aplicación del algoritmo en la aplicación

Gracias a las librerías de OpenCv se puede aplicar el método de Lucas-Kanade de una forma rápida y sencilla.

Una vez obtenidos los puntos singulares o esquinas y haberlos almacenado en una variable apropiada para la función calcOpticalFlowPyrLK(), se procede a llamar a esta.

```
calcOpticalFlowPyrLK(
        prevImg,
        nextImg,
        prevPts,
        nextPts,
        status,
        err,
        winSize,
        maxLevel,
        criteria,
        0,
        0.001);
```

Los parámetros prevImg y nextImg son las imágenes originales. Las siguientes variables son prevPts y nextPts que son los vectores que contienen los puntos singulares en la primera y segunda imagen respectivamente; por consiguiente el primero es de entrada y el segundo es de salida. Los siguientes dos vectores, status y err, aportan información del estado de los puntos singulares, es decir, si ha habido algún problema encontrando el desplazamiento de los puntos.

Por otra parte están las variables winSize, maxLevel, criteria, flags y minEigThreshold. La primera, winSize (winSize=(21,21)), corresponde al tamaño de la ventana de búsqueda en cada nivel de la pirámide y la segunda, maxLevel (maxLevel=3), corresponde al número de pirámides usadas.

A continuación está el parámetro criteria que representa el criterio iterativo de búsqueda del algoritmo y que está configurado de la siguiente forma:

$$\text{criteria.epsilon} = .03;$$

$$\text{criteria.maxCount} = 20;$$

Finalmente, están los parámetros flags y minEigThreshold con valores de 0 y 0.001. Flags permite corregir el posible error surgido y minEigThreshold filtrar los puntos erróneos.

4.4. Análisis de los vectores de movimiento

Obtenidos los puntos singulares o esquinas y su posición en la segunda imagen, ahora es el momento de analizar los vectores de movimiento y clasificarlos.

En primer lugar, se hace una clasificación dependiendo del cuadrante en el que estén. Sólo son de interés aquellos vectores que se encuentren en el primer cuadrante, por lo que, teniendo en cuenta el sentido de los ejes de coordenadas, quedan las siguientes expresiones:

$$nextPts.x > prevPts.x \qquad \&\& \qquad prevPts.y > nextPts.y$$

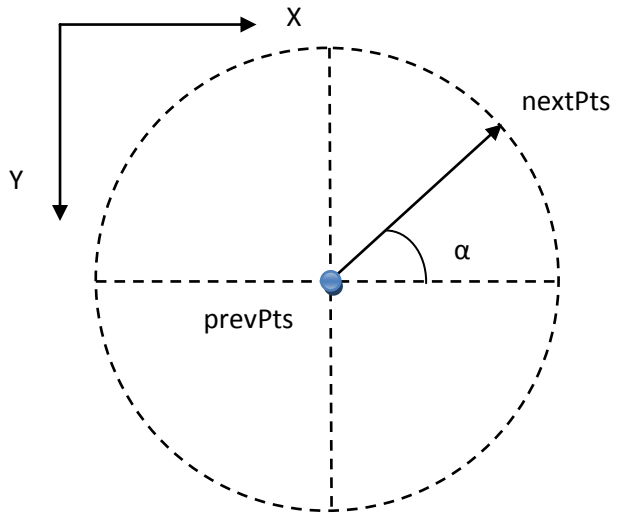

Figura 4.3. Esquema de dirección de los vectores de movimiento

Tras esta primera criba, es menester centrarse en los vectores denominados "GoodVectors" que son aquellos que han pasado el primer corte.

A continuación, se calcula el ángulo α y la hipotenusa de cada uno de los "GoodVectors" y se hace una criba primero teniendo como referencia la hipotenusa y luego el ángulo.

Tras varias pruebas, la hipotenusa debía ser mayor que 3.0 unidades. Puede resultar un valor bajo para la media resultante en la principal prueba (11.33 unidades), pero se dio prioridad a tener falsos positivos antes que a no detectar el adelantamiento, pues como se explicará más adelante, la tenencia de falsos positivos es subsanable.

Siguiendo con los vectores supervivientes, se procede a hacer un filtrado por ángulo, de tal forma que solo aquellos vectores cuyo ángulo esté comprendido entre 6.0° y 55.0° son correctos.

Resumiendo, los "GoodVectors" que permiten detectar un adelantamiento son aquellos que cumplen estas dos condiciones:

$$hipotenusa > 3.0 \qquad \&\& \qquad 6.0° < \alpha < 55.0°$$

4.5. Determinación de la existencia de adelantamiento

Una vez filtrados los "GoodVectors" por su hipotenusa y por su ángulo, ahora es el momento de dilucidar si existe un adelantamiento.

El procedimiento es el siguiente:

1. Se comprueba la siguiente condición:

   $$(\text{left}_{\text{numGoodVectors}} \geq 0.65 * \text{total}_{\text{numGoodVectors}}) \;||\; (\text{left}_{\text{numGoodVectors}} \geq 3)$$

   siendo $\text{left}_{\text{numGoodVectors}}$ el número de vectores que quedan después de las 3 cribas y $\text{total}_{\text{numGoodVectors}}$ el número de vectores después de la primera criba (después de los cuadrantes).

   La primera parte de la condición surgió de la observación de adelantamientos a alta velocidad en los que únicamente había 3 puntos de interés y 2 de ellos indicaban el adelantamiento.

   La segunda parte de la condición se consiguió a partir de la observación de las pruebas realizadas.

2. A continuación, se suma $\text{left}_{\text{numGoodVectors}}$ a la variable $posible\_overtake$ y se comprueba si esta es igual o mayor que 10, de tal forma que:

   $$possible\_overtake = possible\_overtake + \text{left}_{\text{numGoodVectors}}$$
   $$and$$
   $$if\,(possible\_overtake \geq 10)$$

   La variable $possible\_overtake$ representa el número de vectores de movimiento recogidos durante varios instantes, sabiendo que un instante es el par prevImg/ nextImg. De tal forma que para algunos adelantamientos únicamente se requieran varios instantes y otros más claros, solo necesiten un instante.

   Si la condición se verifica, ya hay suficientes indicios para poder dar por bueno el adelantamiento.

En caso de que la condición del paso 1 no se cumpla, se aumenta en uno la variable $non\_possible\_overtake$ y se comprueba su número de forma que si es igual que 7 se descarta el adelantamiento y se reinicia la variable $possible\_overtake$.

Capítulo 5
Resultados obtenidos

El objetivo principal de la aplicación era la detección de adelantamientos en autovía en línea recta para lo cual se buscó un escenario adecuado, como la M30, siendo esta la prueba principal. Sin embargo, en aras de ver hasta donde llegaba la capacidad de la aplicación, también se probó en otros escenarios a priori no tan favorables como una carretera con muchas curvas, una ciudad e incluso un escenario nocturno.

Para poder hacer las pruebas se han utilizado vídeos de youtube y un monitor auxiliar, de tal forma que mientras se reproducía el vídeo en el monitor auxiliar, en el ordenador se iban monitorizando las lecturas provenientes de la tablet que se ubicó justo en frente del monitor auxiliar encima de un soporte construido para las pruebas que permitía sostener la tablet y mantenerla en la posición adecuada sin necesidad de tenerla sujetada.

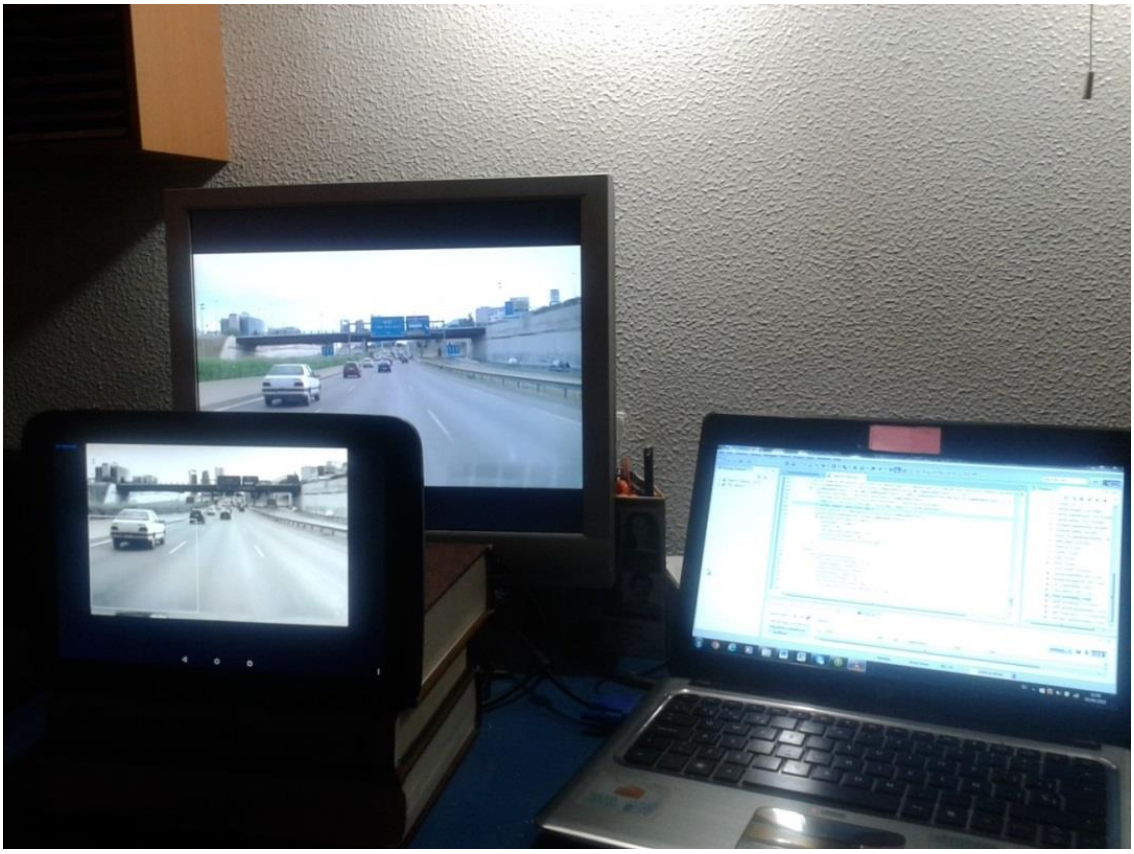

Figura 5.1. Zona de pruebas

En total se han analizado 6 escenarios. En cada escenario se han hecho 5 repeticiones en las cuales se ha analizado si existía adelantamiento en cada caso y en 5 de los 6 escenarios también se ha hecho un análisis de los vectores de movimiento. Todos los datos finales están plasmados en las tablas del anexo.

5.1. M30 de Madrid (Conducción diurna)

Como se ha comentado previamente, este es el escenario principal de pruebas. En este escenario se analizaron casos de vehículos de todas las formas y colores, a distintas velocidades de circulación, incluso en casos de especial dificultad.

Destacan los siguientes casos de gran complejidad:

El caso de un coche que adelanta a alta velocidad y que inmediatamente se cambia de carril a la derecha sin apenas tiempo para detectar el adelantamiento.

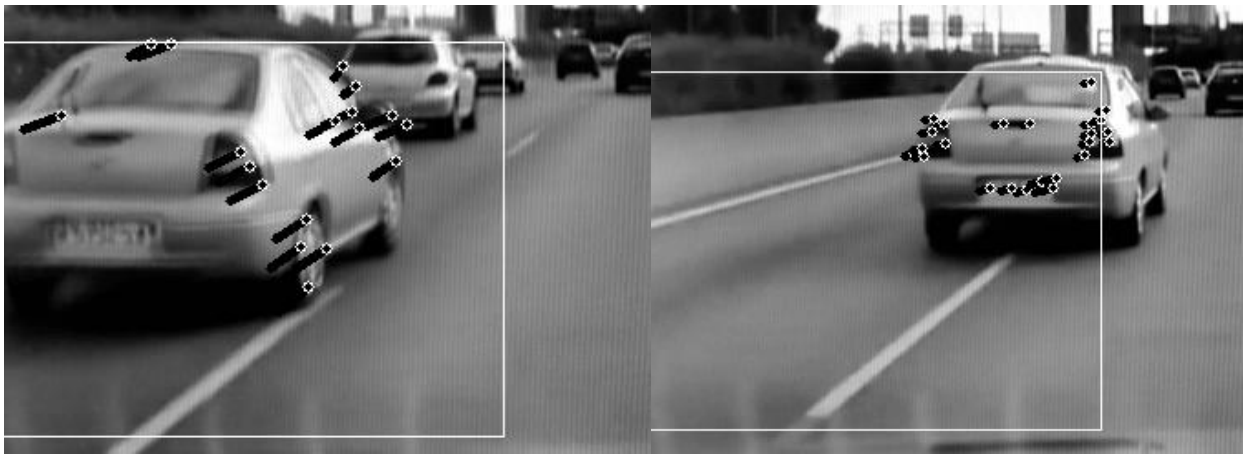

Figura 5.2. M30. Caso de complejidad: un vehículo adelanta a alta velocidad y se cambia de carril inmediatamente

También, en este escenario hay motos:

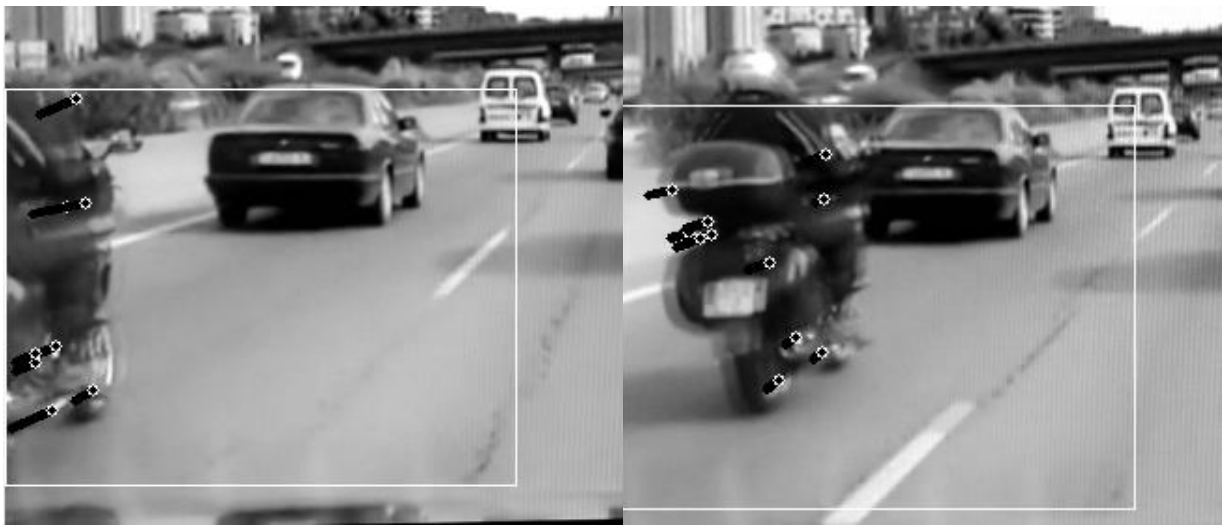

Figura 5.3. M30. Caso de complejidad: una moto

Por otra parte, el caso de un vehículo que adelanta en una zona oscura, es decir, bajo un puente donde apenas hay iluminación.

Figura 5.4. M30. Caso de complejidad: un vehículo adelanta debajo de un puente (una zona oscura)

Finalmente, cabe destacar la existencia de curvas a derecha e izquierda durante el recorrido.

En este test se analizaron 3 tipos de casos:

Posibles falsos positivos	**13**
Casos de vehículos	**32**
Casos totales	**44**

Tabla 5.1. M30: tipos de casos

Dando lugar a los siguientes resultados:

Tipo de error	Tests					Resultado Medio
Falso positivo	3	3	3	2	5	**3,2**
Porcentaje de acierto (%)	76,92	76,92	76,92	84,62	61,54	**75,38**
Coche no detectado	0	0	0	0	0	**0**
Porcentaje de acierto (%)	100	100	100	100	100	**100**
Errores totales	3	3	3	2	5	**3,2**
Porcentaje de acierto (%)	93,18	93,18	93,18	95,45	88,64	**92,73**

Tabla 5.2. M30: resultados

Los resultados obtenidos son bastante aceptables, a pesar de los falsos positivos, que en este escenario son principalmente curvas.

Para más información sobre los casos del escenario y sus resultados, consultar la tabla A1 del Anexo.

5.2. Autobahn 6 de Alemania (Nublado)

Siguiendo en la línea de la M30, este escenario es una carretera alemana de tipo autovía con 2 carriles. La principal diferencia con la M30 es que la ubicación de la cámara es distinta pues en este caso el vehículo es un camión, además de que el cielo está nublado disminuyendo la iluminación.

En este caso se han analizado 3 tipos de casos:

Posibles falsos positivos	**8**
Casos de vehículos	**19**
Casos totales	**27**

Tabla 5.3. Autobahn 6 de Alemania: tipos de casos

Dando lugar a los siguientes resultados:

Tipo de error	Tests					Resultado Medio
Falso positivo	6	4	5	5	6	**5,2**
Porcentaje de acierto (%)	25	50	37,50	37,50	25	**35**
Coche no detectado	0	0	0	0	0	**0**
Porcentaje de acierto (%)	100	100	100	100	100	**100**
Errores totales	6	4	5	5	6	**5,2**
Porcentaje de acierto (%)	77,78	85,19	81,48	81,48	77,78	**80,74**

Tabla 5.4. Autobahn 6 de Alemania: resultados

Por una parte, la detección de vehículos continúa siendo del 100%, por tanto, el factor nublado en principio no afecta.

Por otra parte, como se observa el porcentaje de acierto total ha disminuido. Esto es debido a que los casos de falsos positivos han aumentado. Sin embargo, en este caso no son debidos a curvas, sino que se deben a la detección del lateral, que es la detección en línea recta del arcén. Este tipo de detecciones solo se dan en las curvas, pero si el arcén tiene una mediana se produce este tipo de falsos positivos. En el caso anterior, ya fuera por la calidad de la imagen o por la ubicación de la cámara, no hay detecciones del lateral, salvo en una ocasión en la que el vehículo está en el carril izquierdo; sin embargo en este escenario, a pesar de estar el camión en el carril derecho, hay detecciones del lateral.

Figura 5.5. Autobahn 6. Falso positivo: detección del lateral

Para más información sobre los casos del escenario y sus resultados, consultar la tabla A2 del Anexo.

5.3. Autobahn 9 de Alemania (Atardecer y 3 carriles)

Si en el anterior escenario se disminuía la iluminación ya que estaba nublado, en este escenario se aumenta la iluminación debido al atardecer. Por otra parte, este escenario permite ver si la aplicación puede detectar el tercer el carril (no solo el carril contiguo).

Por tanto, se analizaron los siguientes casos:

Posibles falsos positivos	**4**
Casos de vehículos	**9**
Casos de vehículos en el 3º Carril	**9**
Casos totales	**22**

Tabla 5.5. Autobahn 9 de Alemania: tipos de casos

Dando lugar a los siguientes resultados:

Tipo de error	Tests					Resultado Medio
Falso positivo	3	0	1	1	0	**1**
Porcentaje de acierto (%)	25	100	75	75	100	**75**
Coche no detectado	0	0	0	0	0	**0**
Porcentaje de acierto (%)	100	100	100	100	100	**100**
Casos de vehículos en el 3º Carril	9	9	9	9	9	**9,00**
Porcentaje de acierto (%)	0	0	0	0	0	**0**
Errores totales	12	9	10	10	9	**10**
Porcentaje de acierto (%)	45,45	59,09	54,55	54,55	59,09	**54,55**

Tabla 5.6. Autobahn 9 de Alemania: resultados

Como en el caso anterior, a pesar de que la iluminación ha aumentado debido al atardecer, las detecciones de coches siguen manteniéndose al 100%.

Sin embargo, a pesar del dato anteriormente mencionado, el porcentaje de acierto total ha disminuido considerablemente pues la detección de vehículos en el 3º Carril es del 100%. El resto de falsos positivos (una detección del lateral, dos curvas y un cambio de carril), a penas bajan el porcentaje de acierto total a un 92,31% (si no se tuviera en cuenta los casos del 3º Carril), por lo que se puede concluir que el problema en este escenario es la detección del 3º Carril y no la detección de falsos positivos normales.

Figura 5.6. Autobahn 9. Falso positivo: detección del 3º Carril

Para más información sobre los casos del escenario y sus resultados, consultar la tabla A3 del Anexo.

5.4. Almería (Curvas)

Probado que la detección de vehículos es inmejorable (100%), se pasa a analizar en profundidad las carencias del sistema. En este escenario no se analiza si la iluminación afecta a la detección de vehículos, sino que se analiza el factor de las curvas que es la mayor fuente de falsos positivos en el escenario principal, la M30.

Por tanto, en este escenario se han analizado los siguientes casos:

Posibles falsos positivos	**16**
Casos de vehículos	**2**
Casos totales	**18**

Tabla 5.7. Almería: tipos de casos

Dando lugar a los siguientes resultados:

Tipo de error	Tests					Resultado Medio
Falso positivo	7	8	2	5	5	**5,4**
Porcentaje de acierto (%)	56,25	50,00	87,50	68,75	68,75	**66,25**
Coche no detectado	0	0	0	0	0	**0**
Porcentaje de acierto (%)	100	100	100	100	100	**100,00**
Errores totales	7	8	2	5	5	**5,4**
Porcentaje de acierto (%)	61,11	55,56	88,89	72,22	72,22	**70,00**

Tabla 5.8. Almería: resultados

Desgranado los falsos positivos, el escenario cuenta con:

- 2 cambios de carril.
- 1 detección del lateral.
- 13 Curvas:
    - 6 Curvas a la derecha
    - 7 Curvas a la izquierda

Teniendo en cuenta que los cambios de carriles no han dado lugar a falsos positivos, se vuelve a fallar en los mismos casos de falsos positivos de los escenarios anteriores: las curvas y el lateral.

Figura 5.7. Almería. Falso positivo: detección del lateral

En este caso, las curvas son el principal problema. Si se analizan todas las detecciones de los 5 test en los casos de las curvas, se obtienen 65 casos, de los cuales 18 han dado lugar a un falso positivo (casi un 28%).

Si se analizan dependiendo del sentido de la curva, hay 35 casos de curvas a la izquierda de los cuales han dado positivo 11 de ellos (casi un 31,5%), mientras que hay 30 casos de curvas a la derecha de los cuales han dado positivo 7 de ellos (casi un 23,4%).

Sin embargo, aunque a priori los casos de curvas a la izquierda dan lugar a un mayor número de casos positivos, cabe destacar que uno de los casos de estudio de curva a la izquierda va mezclado con un cambio de carril a la izquierda lo que puede influir considerablemente en la medida. En este caso, de 5 tests, se ha obtenido un falso positivo en 4 de ellos, lo que es bastante extraño porque el resto de curvas como mucho han dado lugar a tres positivos. Si se elimina este caso, el porcentaje de falso positivo en las curvas a la izquierda es del 20%

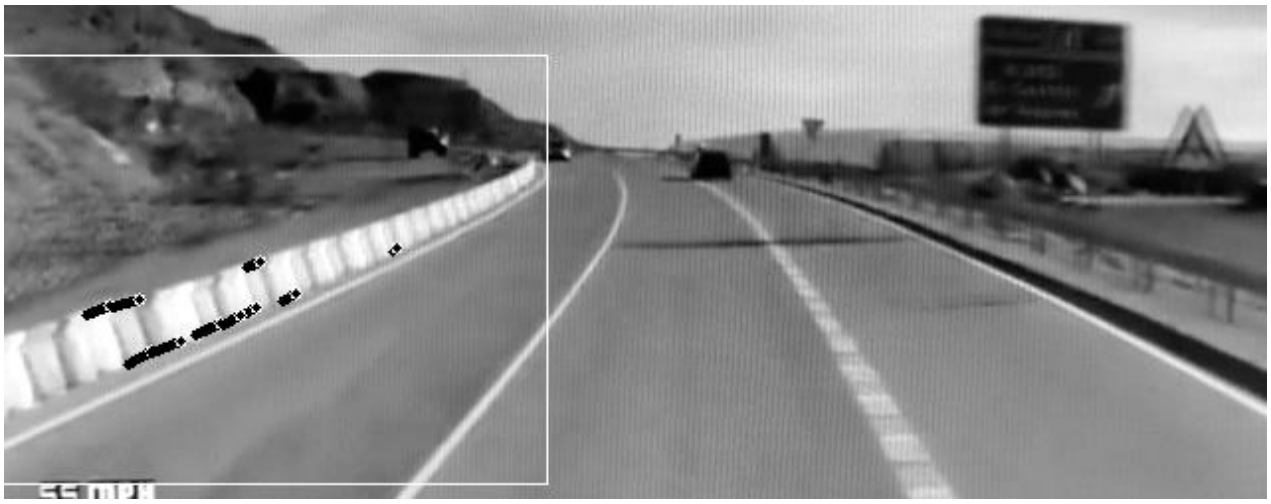

Figura 5.8. Almería. Falso positivo: curva a la izquierda

Para más información sobre los casos del escenario y sus resultados, consultar la tabla A4 del Anexo.

5.5. Majadahonda (Ciudad)

En este escenario, se dejan atrás la carreteras para dar paso a una ciudad, en la cual se presentan casos de resaltos, rotondas y frenadas, es decir, un escenario más complejo.

Estos son los casos que se han analizado:

Posibles falsos positivos	**17**
Falsos positivos con coches (Rotonda)	**2**
Casos de vehículos	**10**
Casos totales	**29**

Tabla 5.9. Majadahonda: tipos de casos

Dando lugar a los siguientes resultados:

Tipo de error	Tests					Resultado Medio
Falso positivo	1	1	3	0	2	**1,4**
Porcentaje de acierto (%)	94,12	94,12	82,35	100	88,24	**91,76**
Falso positivo con coche (Rotonda)	0	0	0	0	0	**0**
Porcentaje de acierto (%)	100	100	100	100	100	**100,00**
Coche no detectado	1	1	2	1	1	**1,2**
Porcentaje de acierto (%)	90	90	80	90	90	**88,00**
Errores totales	2	2	5	1	3	**2,6**
Porcentaje de acierto (%)	93,10	93,10	82,76	96,55	89,66	**91,03**

Tabla 5.10. Majadahonda: resultados

En este escenario, se presentan dos casos de falsos positivos con vehículo implicado, ambos resueltos satisfactoriamente.

En cuanto a los casos de falsos positivos generales, cabe destacar los casos de las rotondas que anteriormente no se habían presentado. En general, la aplicación se ha comportado de forma correcta pues solo ha tenido problemas en el trascurso de una rotonda y en la salida de otra. De los 55 casos (11 casos x 5 tests), han dado positivo 4 de ellos: 3 en el transcurso de la rotonda y 1 a la salida de la rotonda.

Figura 5.9. Majadahonda. Falso positivo: rotonda

Sin embargo, a pesar de la "efectividad" en las rotondas, en este escenario, el número de casos de coches detectados no es del 100%, sino que es del 88%. Esta no detección se ha dado en 2 casos y especialmente en uno de ellos que no se ha detectado en ninguno de los 5 tests.

Este caso consiste en una salida de rotonda en la cual va un coche en paralelo al vehículo de la cámara y sale en la misma salida que el vehículo de la cámara cruzándose y produciendo el adelantamiento. Técnicamente, a pesar, de que no es un adelantamiento ordinario, sí debería haberlo detectado. Sin embargo, nunca lo hace. El motivo principal es la velocidad y el ángulo de movimiento puesto que las líneas de movimiento son pequeñas y muy horizontales.

Figura 5.10. Majadahonda. Coche no detectado: salida de la rotonda

El otro caso, apenas repercute pues no es un adelantamiento como tal, es un caso en el cual hay un vehículo delante de la cámara semiparado y que arranca despacio. Depende de cómo se tome este caso, incluso podría llegar a tomarse como un caso de falso positivo.

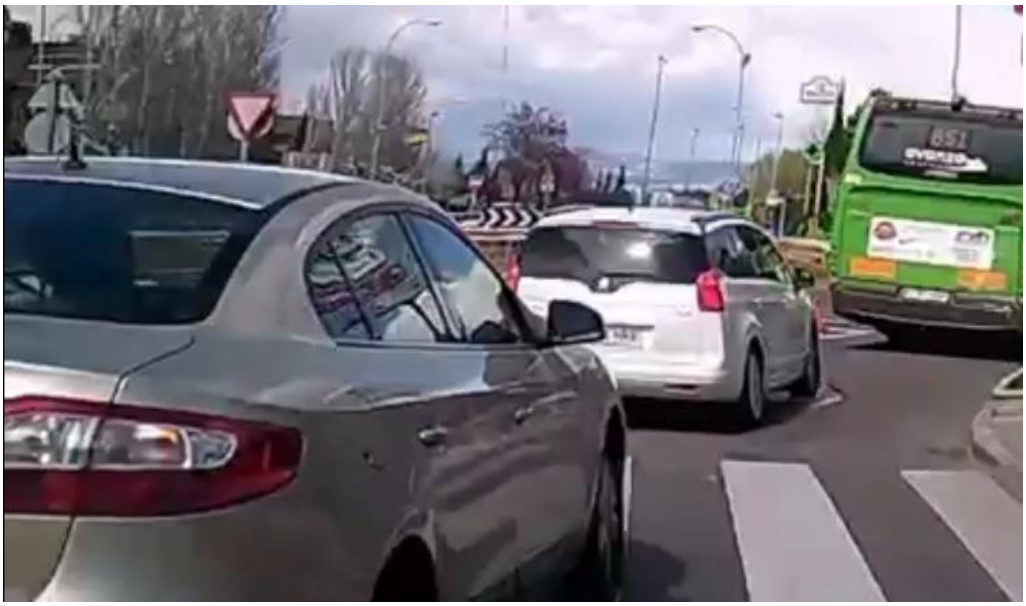

Figura 5.11. Majadahonda. Coche no detectado: vehículo semiparado

Para más información sobre los casos del escenario y sus resultados, consultar la tabla A5 del Anexo.

5.6. Holanda (Conducción nocturna)

Finalmente, tras haber probado la aplicación en diversos escenarios con diferentes características, el último escenario es una autovía de Holanda. Este tipo de escenario ya se ha probado previamente, pero este caso difiere de los anteriores en que en este la iluminación es nula, debido a que es de noche.

A pesar de la nula iluminación, tras varias pruebas cambiando el brillo y el contraste para ver las respuestas de la aplicación, se decidió no tocar ninguna de estas características, pues si se tocaban se dejaban de detectar o bien los coches oscuros o los claros, y con la configuración de brillo y contraste de los anteriores escenarios se consiguió un equilibrio.

Estos son los casos que se han analizado:

Posibles falsos positivos	**17**
Casos de vehículos	**10**
Casos totales	**29**

Tabla 5.11. Holanda: tipos de casos

Dando lugar a los siguientes resultados:

Tipo de error	Tests					Resultado Medio
Falso positivo	4	3	1	1	2	**2,2**
Porcentaje de acierto (%)	33,33	50,00	83,33	83,33	66,67	**63,33**
Coche no detectado	2	3	16	3	2	**5,2**
Porcentaje de acierto (%)	93,94	90,91	51,52	90,91	93,94	**84,24**
Errores totales	6	6	17	4	4	**7,4**
Porcentaje de acierto (%)	84,62	84,62	56,41	89,74	89,74	**81,03**

Tabla 5.12. Holanda: resultados

En este escenario los casos de falsos positivos consisten en curvas y cambios de carriles, siendo los más detectados dos curva a la derecha.

Sin embargo, lo más preocupante de este escenario no son los falsos positivos, sino el "bajo" porcentaje de detecciones de vehículos. En anteriores escenario de autovía este porcentaje era el 100%, pero en este caso el porcentaje baja hasta un 84,24%, por lo que se puede concluir que una baja iluminación, sí que afecta a la detección de vehículos, puesto que el problema no era que los vectores de movimiento no estaban dentro de los parámetros, sino que el algoritmo de obtención de puntos de interés apenas detectaba puntos.

La mayoría de vehículos no detectados únicamente no han sido detectados una vez de cinco posibles, pero hay otros casos en los que los vehículos no han sido detectados varias veces. Hay un caso en especial, que en ninguno de los 5 tests ha sido detectado; este vehículo es oscuro y va a una velocidad alta, además de que enciende las luces traseras y apenas se distingue la silueta del vehículo.

Figura 5.12. Holanda. Coche no detectado

Analizando el resto de no detecciones, hay coches claros y oscuros a distintas velocidades, por lo que el factor determinante es la iluminación.

Sin embargo, sorprende el salto de no detecciones entre los tests 1, 2, 4 y 5 cuya media es de 2,5 vehículos y el test 3 cuyo número de no detecciones es de 16 vehículos. Esta gran diferencia es llamativa, y lleva a pensar que este escenario es cuestión necesita un análisis mayor. Por otra parte, la razón más plausible a esta diferencia es que durante el test 3, la tablet no estaba perfectamente colocada y por tanto en un escenario tan complejo, al no estar la aplicación al 100% (en colocación y en procesamiento) se puede haber "contaminado" el test, dando lugar a estos resultados.

Para más información sobre los casos del escenario y sus resultados, consultar la tabla A6 del Anexo.

5.7. Análisis de los vectores de movimiento como solución a los falsos positivos

A parte de los porcentajes de acierto en la detección de los adelantamientos, también se ha realizado un estudio sobre la longitud y el ángulo de los vectores de movimiento.

Aunque en un principio se creía que los rangos de longitud y de ángulo iban a poder dar solución a los falsos positivos, no fue así. Si se observan los datos medios de las tablas del anexo, los falsos positivos tienen valores de todo tipo e incluso algunos bastante similares a detecciones de vehículos, por lo que se creyó conveniente obtener un porcentaje de detección del 100%, a coste de falsos positivos.

Escenario		Módulo Medio	Ángulo Medio	Módulo Diferencia	Ángulo Diferencia
M30	Casos de Vehículos	11,33	21,99	-3,92	-6,72
	Falsos Positivos	15,25	28,71		
Autobahn 6	Casos de Vehículos	14,47	21,68	1,47	1,81
	Falsos Positivos	13,00	19,87		
Autobahn 9	Casos de Vehículos	19,85	20,81	4,43	4,74
	Falsos Positivos	15,42	16,07		
Almería	Casos de Vehículos	21,51	23,69	9,76	1,57
	Falsos Positivos	11,75	22,12		
Majadahonda	Casos de Vehículos	12,5	20,85	4,47	-2,3
	Falsos Positivos	8,03	23,15		

Tabla 5.13. Resultados del módulo y ángulo medio de los vectores de movimiento

Haciendo un análisis más detallado de los distintos escenarios, se puede ver que la diferencia entre los valores medios de las detecciones y de los falsos positivos es apenas de 10 unidades, lo que implica que no se puede mejorar el porcentaje de falsos positivos cambiando los rangos de módulo y ángulo, sin perder detecciones de vehículos.

Capítulo 6
Conclusiones y trabajos futuros

El objetivo principal del trabajo era explorar la posibilidad del uso de una aplicación Android para detectar los adelantamientos y de esta forma minimizar los accidentes por colisiones laterales.

Desde un principio esta aplicación estuvo pensada para funcionar correctamente únicamente en las líneas rectas de las autovías (conducción diurna), puesto que este trabajo era un estudio de viabilidad. Sin embargo, en aras de ver los límites de la aplicación, se probó en escenarios y en situaciones en las que no se pretendía que funcionara.

Por tanto, una vez analizados los datos de las pruebas, se puede concluir que aunque la aplicación no es perfecta debido a los falsos positivos y a los problemas surgidos en el caso de la conducción nocturna, esta aplicación tiene posibles aplicaciones futuras. Los buenos datos en cuanto al porcentaje de acierto en detección de adelantamientos, hacen ver que esta línea de investigación tiene futuro y que no hay que descartar al sistema Android como vía para la mejora de los sistemas de seguridad en la conducción.

En cuanto a los posibles trabajos futuros se abren varias vías de trabajo. Durante las pruebas se detectaron falsos positivos de tres clases principalmente: las curvas, la detección del lateral y el 3°carril.

Para las curvas y la detección del lateral, se puede integrar un algoritmo para detectar la presencia de un vehículo de forma que únicamente se dé por válido un adelantamiento si hay vectores de movimiento adecuados y existe la presencia de un vehículo. Un posible algoritmo útil para esta tarea podría ser el algoritmo Haar-like en cascada. Este algoritmo permite dilucidar la posibilidad de la presencia de un objeto en particular en cierta región de una imagen.

Por otra parte, este nuevo algoritmo no solucionaría el problema de la detección en el 3°carril, por lo que habría que seguir haciendo modificaciones. La posibilidad más plausible sería la de reducir la zona de detección de forma que solo se coja el carril contiguo. Una forma de hacerlo sería adaptando una aplicación que detecte las líneas de la carretera. El objetivo de esta aplicación es la de evitar que el vehículo se salga del carril, por lo que se podría adaptar para que detecte las líneas del carril contiguo y de esta forma obtener una región de la imagen óptima.

Finalmente, una vez mejorada la aplicación, un trabajo futuro podría ser la instalación del dispositivo Android en un vehículo y si los resultados de las pruebas demuestran que el sistema es viable, integrar el dispositivo al vehículo, de forma que determine si hay un adelantamiento cuando se active el intermitente.

Capítulo 7
Presupuesto

7.1. Coste del material

Concepto	Cantidad	Coste unitario	Coste total
Tablet Nexus 10	1	499,99 €	499,99 €
Ordenado HP	1	399,99 €	399,99 €
Monitor Auxiliar	1	67,05 €	67,05 €
TOTAL			967,03 €

7.2. Coste del personal

Concepto	Cantidad	Coste unitario	Coste total
Graduado en Ingeniería Electrónica Industrial y Automática	6 meses	1600 €/mes	9.600 €
TOTAL			9.600,00 €

7.3. Presupuesto total

Concepto	Coste total
Coste del material	967,03 €
Coste del personal	9.600 €
TOTAL	**10.567,03 €**

Capítulo 8
Bibliografía

[1] DGT: Anuario Estadístico de accidentes 2013
Consultada el 12 de agosto del 2015 en
http://www.dgt.es/Galerias/seguridad-vial/estadisticas-e-indicadores/publicaciones/anuario-estadistico-de-accidentes/anuario-accidentes-2013.pdf

[2] El android libre
Consultada el 12 de agosto del 2015 en
http://www.elandroidelibre.com/2014/01/android-alcanza-los-dos-mil-millones-de-dispositivos-casi-mil-millones-solo-en-2013.html

[3] Xataka: apasionados por la tecnología
Consultada el 13 de agosto en
http://www.xataka.com/automovil/sistemas-de-deteccion-en-los-coches-para-evitar-accidentes

[4] Revista CESVIMAP
Consultada el 13 de agosto en
http://www.revistacesvimap.com/las-nuevas-tecnologias-aplicadas-al-automovil-pueden-evitar-accidentes/

[5] DiarioMotor
Consultada el 13 de agosto en
http://www.diariomotor.com/tecmovia/2013/01/26/mercedes-real-life-assist-evitando-circular-en-direccion-opuesta-laboratorio-tecmovia/

[6] smartGSM
Consultada el 14 de agosto en
http://www.smart-gsm.com/moviles/samsung-google-nexus-10

[7] Wikipedia
Consultada el 14 de agosto
https://es.wikipedia.org/wiki/Eclipse_(software)

[8] Eclipse: marketplace
Consultada el 14 de agosto
https://marketplace.eclipse.org/content/android-development-tools-eclipse

[9] Wikipedia
Consultada el 14 de agosto
https://en.wikipedia.org/wiki/OpenCV

[10] Wikipedia
Consultada el 14 de agosto
https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)

[11] OpenCV
Consultada el 17 de agosto
http://docs.opencv.org/master/dd/d1a/group__imgproc__feature.html#ga1d6bb77486c8f92d79c8793ad995d541

[12] OpenCV
Consultada el 17 de agosto
http://docs.opencv.org/master/d7/d8b/tutorial_py_lucas_kanade.html#gsc.tab=0

[13] Developers Android
Consultada el 19 de agosto
http://developer.android.com/sdk/installing/index.html?pkg=adt

[14] AprendeAndroid
Consultada el 3 de septiembre
http://www.aprendeandroid.com/l1/conectar_movil_eclipse.htm

[15] OpenCv
Consultada el 3 de septiembre
http://docs.opencv.org/java/2.4.9/

[16] Youtube
Consultada el 3 de septiembre
https://www.youtube.com/watch?v=-KShPTHo3-0

[17] Youtube
Consultada el 3 de septiembre
https://www.youtube.com/watch?v=QWW6J20sd9M

[18] Youtube
Consultada el 3 de septiembre
https://www.youtube.com/watch?v=q9n49W6inMw

[19] Youtube
Consultada el 3 de septiembre
https://www.youtube.com/watch?v=dphETHhnBRs

[20] Youtube
Consultada el 3 de septiembre
https://www.youtube.com/watch?v=4NoGz1sjdtc

[21] Youtube
Consultada el 3 de septiembre
https://www.youtube.com/watch?v=bdmLFJW_t-k

[22] Circula Seguro
Consultada el 13 de agosto
http://img.blogs.es/circulaseguro/wp-content/uploads/2014/02/Detector-fatiga-sue%C3%B1o-volante.jpg

Anexos

Tabla A0: Leyenda de las pruebas

Columna	Color	Significado
Caso		Debería detectarse
Caso		NO debería detectarse
Detección de Adelantamiento		El evento se ha producido de forma correcta
Detección de Adelantamiento		Se ha detectado adelantamiento cuando no debería haber sido detectado
Detección de Adelantamiento		El vehículo no ha sido detectado
Medidas Medias (Módulo y Ángulo)		El evento se ha producido de forma correcta
Medidas Medias (Módulo y Ángulo)		1 Error (Falso positivo)
Medidas Medias (Módulo y Ángulo)		2 Errores (Falso positivo)
Medidas Medias (Módulo y Ángulo)		3 Errores (Falso positivo)
Medidas Medias (Módulo y Ángulo)		4 o 5 Errores (Falso positivo)
Medidas Medias (Módulo y Ángulo)		Entre 1 y 4 Errores (Vehículo NO detectado)
Medidas Medias (Módulo y Ángulo)		5 Errores (Vehículo NO detectado)

Tabla A1: Carretera M30 de Madrid (Prueba principal)

Carretera M30 de Madrid (Soleado)											
Caso	Tiempo	Color	Velocidad	Detección de Adelantamiento					Módulo Medio	Ángulo Medio
	0:00:01	Rojo	Media						4,62	19,18
	0:00:12	Claro	Media						17,19	25,21
	0:00:20	Claro	Media-Baja						6,00	33,45
Un coche adelanta y se cambia al carril de la derecha	0:00:28	Claro	Alta						6,96	16,74
Curva Izquierda	0:00:34	Sin coche	-						-	-
	0:00:49	Oscuro	Media						9,48	19,10
	0:00:56	Claro	Media						8,34	24,22
	0:01:04	Oscuro	Media						11,79	11,53
Moto	0:01:07	Oscuro	Alta						8,83	18,27
	0:01:19	Mostaza	Media						19,75	17,43
	0:01:24	Oscuro	Baja						7,04	29,96

	0:01:36	Oscuro	Media						5,10	29,61
Entrada Curva Izquierda	0:01:50	Sin coche	-						-	-
	0:01:54	Rojo	Alta						13,41	18,48
Curva Izquierda	0:02:00	Sin coche	-						6,96	43,65
	0:02:14	Claro	Alta						10,27	19,60
Curva Izquierda	0:02:17	Sin coche	-						-	-
	0:02:24	Rojo Oscuro	Alta						16,33	19,77
Curva Derecha	0:02:52	Sin coche	-							
	0:03:04	Oscuro	Baja						8,12	23,20
Entrada Curva Derecha	0:03:34	Sin coche	-						8,64	12,89
Curva Derecha	0:03:47	Oscuro	Media						7,84	25,16
Salida Curva Derecha	0:03:55	Sin coche	-						7,75	33,98
	0:04:16	Oscuro	Media						8,41	28,36

	0:04:25	Gris	Media						7,67	25,57
	0:04:36	Claro	Media						12,32	19,04
Curva Izquierda	0:04:43	Sin coche	-						-	-
Curva Derecha	0:04:50	Sin coche	-						-	-
Curva Derecha	0:04:57	Oscuro	Media						12,57	17,96
Curva Derecha	0:05:07	Rojo	Media						7,66	29,71
	0:05:30	Azul grisáceo	Alta						24,86	25,06
Zona Oscura	0:05:38	Oscuro	Alta						21,40	22,38
	0:05:43	Blanco	Alta						13,43	18,53
	0:05:47	Oscuro	Alta						23,60	15,71
	0:05:52	Claro	Alta						10,89	17,57
Moto en curva a la izquierda	0:05:56	Oscuro	Media						11,14	21,48
Curva Izquierda	0:05:59	Sin coche	-						6,2	38,89

	0:06:03	Claro	Alta						7,35	28,24
Salida Curva Izquierda	0:06:05	Sin coche	-						5,00	27,29
	0:06:20	Claro	Media						8,78	21,57
	0:06:28	Oscuro	Baja						6,78	21,30
	0:06:40	Oscuro	Media						13,41	18,26
Cambio de carril a la Izquierda	0:06:47	Sin coche	-						34,04	21,69
Curva Derecha mientras el vehículo está en el carril izquierdo	0:06:52	Sin coche	-						38,15	22,58

Tabla A2: Autobahn 6 (Nublado)

Autobahn 6. Alemania (Nublado y camión)											
Caso	Tiempo	Color	Velocidad	Detección de Adelantamiento					Módulo Medio	Ángulo Medio
	0:00:15	Gris Claro	Media-Alta						26,90	23,61
	0:00:18	Gris Claro	Media-Alta						18,05	19,42
	0:00:24	Azul Oscuro	Media-Alta						14,38	13,87
	0:00:26	Gris Claro	Media-Alta						15,14	20,14
Detección del lateral	0:00:33	Sin coche	-						13,05	29,70
	0:00:56	Negro	Media-Alta						11,88	20,47
Detección del lateral	0:01:02	Sin coche	-						11,08	19,24
	0:01:10	Azul Oscuro	Media						15,06	28,64
	0:01:13	Gris Oscuro	Media-Alta						22,51	26,87
La cámara bota	0:01:17	Sin coche	-						-	-
	0:01:21	Gris Claro	Media-Alta						16,72	21,57

	0:01:24	Blanco	Media-Alta						9,80	27,10
	0:01:29	Gris Claro	Media-Alta						9,70	27,95
Detección del lateral	0:01:40	Sin coche	-						4,67	14,50
	0:02:00	Negro	Media						9,97	18,98
	0:02:04	Gris Claro	Media-Alta						10,79	24,89
	0:02:14	Azul Oscuro	Media-Alta						10,78	17,34
Furgoneta Media	0:02:19	Gris Claro	Media-Alta						12,97	18,26
Furgoneta Grande	0:02:22	Blanco	Media-Alta						10,60	21,65
	0:02:28	Negro	Media-Alta						12,06	20,80
Detección del lateral	0:02:40	Sin coche	-						16,25	15,98
Curva a la Izquierda cuesta arriba	0:02:56	Sin coche	-						15,34	18,90
	0:03:14	Negro	Media-Alta						23,80	22,22
Cambio de carril a la derecha	0:03:26	Sin coche	-						-	-

Camión. Curva hacia la derecha cuesta arriba	0:03:47	Blanco	Baja						13,30	18,83
Coche en el 3º carril (No debería detectarse)	0:03:57	Negro	Alta						17,60	20,90
Camión. Cuesta arriba	0:04:08	Azul Oscuro	Baja						10,59	19,27

Tabla A3: Autobahn 9 (3 carriles)

Autobahn 9. Alemania (Atardecer, camión y 3 carriles)											
Caso	Tiempo	Color	Velocidad	Detección de Adelantamiento					Módulo Medio	Ángulo Medio
	0:00:01	Gris Claro	Media-Alta						31,52	23,45
3º Carril	0:00:03	Gris Claro	Media-Alta						8,21	24,51
	0:00:06	Gris Oscuro	Media-Alta						13,90	21,37
Curva a la izquierda	0:00:20	Sin coche	-						8,28	17,95
Detección del lateral	0:00:35	Sin coche	-						8,22	10,66
3º Carril	0:00:40	Gris Oscuro	Alta						22,40	10,88
	0:00:41	Gris Oscuro	Media						18,46	16,93
3º Carril	0:00:44	Gris Oscuro	Alta						14,84	19,18
3º Carril	0:00:52	Gris Oscuro	Alta						19,03	13,33
3º Carril	0:00:53	Gris Oscuro	Alta						13,11	14,98
3º Carril	0:00:55	Gris Oscuro	Alta						12,99	16,43

Adelanta y se cambia al carril de la izquierda	0:00:55	Gris Oscuro	Media						16,46	18,77
Adelanta y se cambia al carril de la izquierda	0:01:04	Gris Oscuro	Media-Alta						24,89	23,00
	0:01:08	Negro	Media-Alta						23,26	28,19
3º Carril	0:01:20	Azul Grisáceo	Alta						27,83	16,48
3º Carril	0:01:22	Negro	Alta						17,33	13,83
	0:01:24	Negro	Media						12,25	13,85
	0:01:29	Negro	Media						23,16	22,86
Curva a la izquierda	0:01:42	Sin coche	-						16,73	12,97
3º Carril	0:01:48	Blanca	Alta						16,02	21,58
Furgoneta	0:01:50	Blanca	Media						14,75	23,20
Cambio de carril del camión al carril de la izquierda	0:02:09	Sin coche	-						-	-

Tabla A4: N340 a Almería (Curvas)

N340 a Almería (Camión. Curvas y soleado)											
Caso	Tiempo	Color	Velocidad	Detección de Adelantamiento					Módulo Medio	Ángulo Medio
Cambio de carril a la izquierda	0:00:01	Sin coche	-						-	-
Cambio de carril a la derecha	0:00:03	Sin coche	-						-	-
	0:00:14	Rojo	Alta						23,12	24,11
Cambio de carril a la izquierda. Curva Izquierda	0:00:20	Sin coche	-						10,72	19,78
Curva Derecha	0:00:46	Sin coche	-						5,94	35,06
Curva Izquierda	0:01:09	Sin coche	-						16,22	31,04
Detección del lateral del puente	0:01:23	Sin coche	-						10,09	17,76
Curva Derecha	0:01:37	Sin coche	-						-	-
Curva Derecha	0:01:45	Oscuro	Media						19,90	23,27
Curva Izquierda	0:01:55	Sin coche	-						11,22	23,48
Curva Derecha	0:02:24	Sin coche	-						6,66	11,94

Curva Izquierda	0:02:55	Sin coche	-						20,4	23,87
Curva Derecha	0:03:04	Sin coche	-						21,68	14,40
Curva Izquierda	0:03:20	Sin coche	-						13,43	17,92
Curva Derecha	0:03:37	Sin coche	-						11,26	15,11
Curva Izquierda	0:03:52	Sin coche	-						-	-
Curva Derecha	0:04:17	Sin coche	-						7,75	30,53
Curva Izquierda	0:04:42	Sin coche	-						5,71	24,55

Tabla A5: Majadahonda (Ciudad)

Majadahonda (Ciudad y Soleado)												
Caso	Tiempo	Color	Velocidad	Detección de Adelantamiento					Módulo Medio	Ángulo Medio
Entrada a la rotonda	0:01:01	Sin coche	-						-	-
Rotonda	0:01:05	Sin coche	-						-	-
Rotonda: Coche delante	0:01:09	Gris Claro	Baja						-	-
Rotonda: un coche se incorpora a esta y se coloca en la zona de detección	0:01:16	Gris Claro	Baja						-	-
Salida de la rotonda	0:01:26	Gris Claro	Baja						17,77	15,11
Adelanta y se cambia al carril de la derecha	0:01:50	Gris Oscuro	Media						19,96	17,26
Entrada a la rotonda	0:02:05	Sin coche	-						-	-
Rotonda	0:02:14	Blanco	Baja						12,01	12,54
Rotonda	0:02:20	Sin coche	-						5,25	21,16
Salida de la rotonda	0:02:26	Blanco	Baja						-	-
Ligera Curva a la izquierda	0:02:30	Sin coche	-						-	-

	0:02:40	Gris Claro	Baja						8,96	15,59
Ligera Curva a la derecha	0:02:55	Sin coche	-						-	-
Resalto. Detección del coche de delante	0:03:10	Blanco	Baja						13,39	31,24
Entrada a la rotonda	0:03:13	Sin coche	-						-	-
Rotonda	0:03:15	Sin coche	-						-	-
Salida de la rotonda	0:03:19	Blanco	Baja						5,95	12,81
Coche de antes. Resalto	0:03:26	Blanco	Baja						11,32	42,99
Resalto	0:03:38	Blanco	Muy Baja						13,35	17,29
Un coche delante semiparado y que arranca despacito.	0:04:00	Gris Claro	Baja						9,83	22,81
Rotonda	0:04:23	Sin coche	-						-	-
Salida de la rotonda	0:04:31	Sin coche	-						-	-
Carril izquierdo	0:04:39	Sin coche	-						9,48	26,82
Cambio de carril a la derecha	0:04:48	Sin coche	-						9,48	16,98

Resalto	0:04:50	Sin coche	-						7,90	27,63
Entrada a la rotonda	0:04:55	Sin coche	-						-	-
Rotonda	0:04:57	Sin coche	-						-	-
Salida de la rotonda	0:05:02	Sin coche	-						-	-
Resalto	0:05:04	Sin coche	-						-	-

Tabla A6: Carretera de Holanda (Noche)

Carretera de Holanda (Noche)									
Caso	Tiempo	Color	Velocidad	Detección de Adelantamiento					
Coche de policía	0:00:50	Claro	Media						
	0:01:02	Claro	Media						
	0:01:15	Oscuro	Media						
Cambio de carril a la dch	0:01:17	Sin coche	-						
	0:01:48	Oscuro	Alta						
Curva Derecha	0:02:03	Sin coche	-						
Cambio de carril a la izquierda	0:02:28	Sin coche	-						
Curva Derecha (Ligera pero con resaltos)	0:02:40	Sin coche	-						
	0:02:54	Claro	Media						
	0:03:02	Oscuro	Media						
	0:03:12	Oscuro	Baja						

	0:03:19	Oscuro	Media					
	0:03:28	Oscuro	Media					
	0:03:40	Oscuro	Alta					
	0:03:45	Oscuro	Alta					
	0:03:50	Oscuro	Media					
	0:03:55	Rojo	Media					
	0:04:00	Claro	Alta					
	0:04:02	Claro	Alta					
	0:04:04	Oscuro	Media					
	0:04:07	Oscuro	Media					
	0:04:12	Claro	Media					
	0:04:17	Oscuro	Media					
	0:04:23	Claro	Alta					

	0:04:25	Oscuro	Media					
	0:04:30	Oscuro	Media					
	0:04:33	Claro	Media					
	0:04:47	Claro	Baja					
El vehículo que adelanta vuelve a ser adelantado	0:04:57	Claro	Baja					
Vehículo anterior	0:05:13	Claro	Muy Baja					
El vehículo que adelanta vuelve a ser adelantado	0:05:29	Claro	Baja					
Curva Derecha	0:06:00	Sin coche	-					
Curva Derecha	0:06:02	Claro	Media					
El coche adelanta y cambia de carril a la derecha	0:06:24	Oscuro	Media					
	0:06:34	Oscuro	Media					
	0:06:41	Oscuro	Alta					
	0:06:45	Claro	Alta					

Curva Izquierda	0:06:52	Sin coche	-					
Curva a la Izq. Adelanta y se cambia al carril de la Drch	0:07:05	Claro	Media					